

2004 International Symposium on
Distributed Computing and Applications to
Business, Engineering and Science

DCABES 2004

PROCEEDINGS

Volume I

Editor in Chief Guo Qingping



Hubei Science and Technology Press, Wuhan, China

**2004 International Symposium on
Distributed Computing and Applications to
Business, Engineering and Science**

DCABES 2004

PROCEEDINGS

Volume I

Editor in Chief Guo Qingping

Wuhan, China

September 13-16, 2004

Hubei Science and Technology Press, Wuhan, China

图书在版编目(CIP)数据

2004 年电子商务、工程及科学领域的分布式计算和应用
国际学术研讨会论文集 / 郭庆平主编. — 武汉: 湖北科
学技术出版社, 2004. 9

ISBN 7-5352-3269-8

I. 2... II. 郭... III. 分布式计算机—计算机应用—
国际学术会议—文集 IV. TP338. 8-53

中国版本图书馆 CIP 数据核字 (2004) 第 093693 号

Copyright © 2004 by Hubei Science and Technology Press, Wuhan, China
All Rights Reserved

Copyright and Reprint Permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy for private use. Instructors are permitted to photocopy for private use isolated articles for non-commercial classroom use without fee. Other copying, reprint, or republication requests should be addressed to: Hubei Science and Technology Press, the 13th Floor of Block B, 268 Xiongchu Avenue, Wuhan, 430070, P. R. China

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the Wuhan University of Technology, the Hubei Science and Technology Press, the Natural Science Foundation of China, the Ministry of Education, China, or other sponsors and organizers.

Organized by

WUT Wuhan University of Technology

Co-organized by

ISTCA International Science and Technology
Cooperation Association of Hubei Province

CAA Computer Academic Association of
Hubei Province & Wuhan Metropolis

Sponsored by

WUT Wuhan University of Technology

MOE Ministry of Education, China

NSFC National Nature Science Foundation of
China

SUN Microsystems (Hong Kong Headquarter)



DCABES 2004 PROCEEDINGS

© Editor in Chief Guo Qingping

Editorial Production by Wu Ruilin, Li Zhiming

Cover Art Production by Da Min Wan Xiaofen

Published by *Hubei Science and Technology Press, Wuhan, China*

Tel: +86-(0)27-87679468

Address: The 13th Floor of Block B, 268 Xiongchu Avenue, Wuhan, Post Code: 430070 P. R. China

Printed in Wuhan, China by Youngster Union Printery

Post Code: 430063

880mm×1230mm sixteenmo 71.5 sheets

1 500 k Characters

First Edition September 2004

First Impression September 2004

ISBN 7-5352-3269-8/ TP·79

Price RMB 150

QoS of Book Printing and Bounding is guaranteed by the printery.

CONTENTS

Preface.....	xvi
Committees.....	xvii
Referees.....	xviii

Volume I

1. Grid Computing.....	1
P-GRADE: a High-level Grid Application Development Environment Peter Kacsuk.....	1
High Performance and Grid Computing Where Are We Going? Simon See.....	7
A Taxonomy and Survey of Grid Resource Planning and Reservation Systems for Grid Enabled Analysis Environment Arshad Ali, Ashiq Anjum, Atif Mehmood, et al.....	13
A Comparative Survey of Fault-tolerant and Load-balanced MPI Implementations, Software Packages and Algorithms Raihan Ur Rasool, Guo Qingping.....	17
Reusing Legacy Applications for Grid Computing Yu Huashan, Xu Zhuoqun, Ding Wenkui.....	23
Researches of Key Technologies for Data Grid Fansong Meng, Zude Zhou, Quan Liu.....	29
Global Grid Queue Services Architecture and Point-based Simulated Annealing Algorithm for Resource Scheduling Shengjun Li, Ruimin Shen, Robert Lackman.....	34
The Application of Grid in Grid Services Zhao Jianmin, Zhu Xinzhong.....	37
A Software Bus Based on the Grid Computing Cheng Yuanbin.....	40
Developing Grid Computing Applications Based on OGSA Jing Tong, Haochun Liu.....	43
Research on the Architecture-based Adaptive Grid Application Guoyou Zhang, Yinzhang Guo.....	48
Grid Computing Resource Management Scheduler Based on Genetic Algorithm Hao Tian, Zhou Zude, Liu Quan.....	51
The Scheme of Synthetic Forces Express Based on Vega Grid Cheng Zhou, Qingping Guo.....	55
Implementing Distributed Simulations in Grid Computing Environments Tingxin Song, Cheng Wang, Jianmin Xiong, et al.....	59

HACP: An Ant-Based Partitioner for Grid Computing Applications Lin Jin, Wang Meiqing, Jiang Xiufeng.....	63
A Service-oriented Grid Computing Model Based on Jini Tang Guosheng, Guo Qingping, Yang Jian.....	68
A Layered Distribute Resource Discovery Model in Grid Computing System Wang Xiaogen, Xu Wenbo.....	72
Economic Mechanism Driven Resource Management in Computational Grid Li Chunlin, Lu Zhengding, Li Layuan.....	75
A Taxonomy of Data Location Themes for Storage Cluster and Storage Grid Yong Feng, Yanyuan Zhang.....	80
Research on the Data Grid Griddaen Architecture Based on Grid Middleware Quan Long, Quan Liu.....	86
The Application of Simulation in Large Scale Traffic Flows System Based on Grid Computing Yun Wu.....	90
2. System Architectures, Networking and Protocols.....	94
A Distributed Peer-to-Peer Platform for Synchronized Group Collaboration and Knowledge Sharing Jo-Yew Tham; Seng-Luan Lee; Choon-Ee Tan, Roger; et al.....	94
The Splitting Methods in High-speed Networks Data Analysis Chen Xunxun, Fang Binxing, Li Lei.....	100
Design and Implementation of Policy-based Network Management Based on SNMPv3 Yi Yue, Debao Xiao.....	104
Implementations of CGI in Embedded Web Server Xihuang Zhang, Wenbo Xu.....	107
Modeling Distributed Systems: Architecture and Process Beihong Jin, Jianchao Wang.....	110
Opening Component Integration Architecture Zhou Xiaofeng, Wang Zhijian, Fei Yukui.....	116
Packet Transfer Delay of the RPR Rings in Comparison between the Store-and-Forward and Cut-Through Architecture Yi Yang, Mingcui Cao, Ping Huang.....	120
A Kind of Generic Real-time Dependable Server Architecture with Low Fault-latency Using COTS Components Ou Zhonghong, Dai Xingfa, Yuan Youguang, et al.....	124
Hardened VPN Based on IXP425 Yue Hu, Fangmin Li, Quan Liu.....	128
A New Distributed CFAR Processor Yang Jun, Ma Xiaoyan, Xiang Jiabin.....	131
A Scalable Approach for IP-Multicast in Differentiated Services Networks Wang Xiaoyan, Zheng Mingchun.....	135
A Virtual Server Scheme Based on LVS and XRN Guo Yucheng, Guo Qingping.....	140

An Improved BP Algorithm Based on a Variant Sigmoid Function with Three Parameters Hu Yaogai, Yan Xin, Zhang Xiaoxing, et al.....	144
Implementing Synchronous Multicasting in Switch-Based Cluster Systems Feng Ping, Lei Yanjing, Liu Junrui.....	149
A Study of Personalized Network Based on Multi-Layer Packet Classification and ACL Cheng Chuanhui, Li Layuan, Xiang Yang.....	154
A Hierarchical System Management Approach Based on SNMP for Network Device Yan Bin, Yang Zijie.....	157
VLAN Aggregation Technology Research and Implementation Cheng Chuanqing.....	161
Study and Application on Time Synchronization Technique in Computer Network He Peng, Xu Yishan, Tao Ke, et al.....	165
A Distributed Computing Platform ----BOINC Fan Yang, Xinzhong Zhu, Jianmin Zhao.....	170
The Application of ACE in Distributed Network Management System Chen Jun, Guo Qingping.....	174
A Link Layer Automatic Topology Discovery Algorithm Liu Yuhua, Yu Shengsheng, Li Yanhong.....	178
The Evaluation of the Probabilistic Packet Marking for Path Bifurcation Fu Jianming, Zhu Qin, Zhang Huanguo.....	182
A Multicast Routing Optimization Algorithm with Bandwidth and Delay Constraints Based on GA Sun Baolin, Li Layuan, Ma Jun.....	186
A Kind of Low Latency Communication Way over Ethernet Dai Xinfu, Ou Zhonghong, Fang Ming, et al.....	192
A QoS Multicast Routing Algorithm Working with Imprecise State Information Yan Xin, Li Layuan, Zhang Xiaoxing.....	196
Data Transmission Rate Control in Computer Networks Using Neural Predictive Networks Liansheng Tan, Naixue Xiong, Yan Yang.....	202
Research on the Basis of QoS Routing Protocol of Ad Hoc Network Chen Niansheng, Li Layuan.....	206
An Improved Multicast Routing Algorithm with Delay-constrained Based on Genetic Algorithm Wei Fang, Wenbo Xu.....	211
The Distributed Interactive Multimedia Synchronization Model Based on the Temporal Petri net Lu Feng, Guo Yingli.....	216
An Improved Genetic Algorithm for Solving QoS Distributed Routing Problem Youwei Yuan, C.Cujaj.....	221
Design and Implementation of an Embedded VPN Gateway Based on IPSec Zheng Yuanjiu, Liu Quan, Li Fangmin.....	225
Shortest-Path Routing Based on Ant-Algorithm Min LianYing, Yang JinYong.....	228
Data Communication between Monitor computer and PLC Based on the Profibus Liu Qing, Guo Jianming, Wang Yanwen.....	231

A Secure QoS Multicast Routing Protocol Yang Mingxi, Li Layuan.....	235
Research of Indirect Replication Algorithm in Distributed Storage System Wang Yijie, Li Sikun.....	241
Design and Performance Analysis of High Availability iSCSI Storage Area Network Jiang Minghua, Zhou Jingli.....	243
A High Performance Dynamic Memory Management Scheme Huifu Zhang, Fangmin Li.....	247
The Performative Design of Distributed Storage Agents Zhu Yong, Zhang Jiangling.....	251
A New Solution Scheme of NAT and IPSec Protocol Compatibility Problem Based on IP Tunnel Li Fangmin, Xue Ligong, Wang Runyun, et al.....	255
3. Mobile Computing.....	258
Some Strategies in the Distributed and Mobile GPS System Min Peng, Yanxiang He, Wensheng Hu	258
Efficient and Adaptive Load Balancing Based on Mobile Agent Yang Yongjian, Chen Yajun, Cao Xiaodong, et al.....	263
A Mobile Agent Based Middleware for Grid Computing Peng Dewei, He Yanxiang.....	268
Research on Prediction Model of Dynamic Load-Balancing with Mobile Agent in a Parallel Distributed System Yang Yongjian, Cao Xiaodong, Chen Yajun.....	274
Comparison of Missing Data Estimation Methods in Satellite Information for Scientific Exploration ZHAO Guanghui, Song HuaZhu, Xia HongXia, et al.....	278
QoS-based Multicast Routing Optimization Algorithms for Wireless Networks Chen Hua, Sun Baolin.....	281
A Core-Stateless Dynamic Bandwidth Allocation Mechanism Based on Resource Reservation Liu Quan, Liang Xiaoyu, Li Fangmin.....	286
Customer Relation Management System Based on Mobile Internet Di Guoqiang.....	290
4. Parallel / Distributed Algorithms.....	293
A Dynamic Data-Driven Application Simulation Framework Craig C. Douglas, Yalchin Efendiev.....	293
A Hybrid Method for the Update of Sub-domain Interfaces Qingping Guo, C.-H. Lai.....	298
On a Distributed Algorithm for the Solution of Nonlinear Transient Parabolic Problems C.-H. Lai, A. J. Davies.....	301
Parallel and Multilevel Algorithms for Computational Partial Differential Equations Peter K Jimack.....	305

A Parallel Asynchronous Hybrid Method to Accelerate Convergence of a Linear System Haiwu He, Guy Bergere, Serge Petiton.....	311
Design and Verification of Parallel Programs Wang Jian, Chi Xuebin.....	317
Protein Evolution Based on Complex Networks Wang zhongjun, Wang nengchao.....	320
Algorithm of Decomposition and Reconstruction with Orthogonal Multiwavelet Packets with Random Scale Leng Jinsong, Huang Tingzhu.....	323
Distributed Cluster-based Solution Techniques for Dense Linear Equations Gu Zhimin, Marta Kwiatkowska.....	326
Apply Neural Computation to Ground Waves Caused by High-Speed Trains Zou Chengming, Yang Hongyun, Tong Qiwei, et al.....	331
A New Distributed Computing Model: Isolated Island Xinchao Zhao.....	334
Algorithm for Solving the Symmetric Five-Diagonal Toeplitz Linear Equations Ran Ruisheng, Huang Tingzhu.....	339
The Fuzzy Inference Based on Genetic Algorithm Zhang Jianhua, Jiang Qian.....	342
Parallel Multi-grid Algorithm Based on Cluster Computing with Application to Transient Heat Transfer Zongbo Zhu, Guoxun Yang, Chunxiao Liu, et al.....	345
An Error Bound for the SAOR Method Liu Futi, Huang Tingzhu, He huiming.....	350
Analysis of Parallel Matrix Multiplication Algorithms Wanlong Liu, Yaolin Gu.....	354
Multi-stage Influence Diagrams Decision Using Genetic Algorithms Zhao Yun, Liu Weiyl, Li Jin.....	358
Genetic Searching for Optimized Closure State of CFST Arch Bridge Construction Fan Jianfeng, Zhong Luo, Tong Qiwei.....	362
A Wavelength Assignment Algorithm of Parallel LU Decomposition Communication Pattern on WDM Ring Interconnection Network Yawen Chen, Fangai Liu.....	366
Genetic Algorithms for Solving Graphical Games Jin Li, Weiyl Liu, Yun Zhao.....	372
Some Issues on Adaptive Genetic Algorithm Zhang Jianjian, Wang Pan.....	377
A Class of Accelerated Convergence Algorithms for Solving Ordinary Differential Systems Dongjin Yuan.....	381
Genetic Algorithm and Evolutionary Programming: A comparison study Wei Gao.....	385
A High-efficient Parallel Reasoning Algorithm Minghe Huang, Cuixiang Zhong.....	390

A Fast and Efficient Parallel Sorting Algorithm on LARPBS Chen Hongjian, Chen Yixin, Chen Ling, et al.....	393
Fast Parallel Identification of Multi-peaks in Function Optimization Guo Guanqi, Tan Zhumei.....	398
5. Computational Methods	
Constraint-based Concurrency in Java Rafael Ramirez, Juanjo Martinez, Andrew E. Santosa.....	402
Component Based Simulation Environments of Distributed Discrete Event Simulation Zhang Yaohong , Luo Xueshan, Luo Aiming, et al.....	407
An Approach of Component Tailoring Based-on Parameterized Contracts Fei Yuikui, Wang Zhijian.....	411
On a Smart Software for Cement-Meal Batching Computation Jiang Hongzhou, Li Juanjuan.....	415
6. Distributed Operating System	
A New Simulation Method Using Multithreading for Modeling Parallel Operated Systems Wingcheong Kwong.....	418
The Design and Implementation of a Multi-Auctioneer Prototype System for Grid Resource Management Xiuchuan Wu, Hao Li, Jiubin Ju.....	424
Research on Resilient Distributed File Systems Li Zhonghua, Li Weihua, Zhang Lin.....	430
Efficient Scheduling of Task Graphs to Multiprocessors Using a Simulated Annealing Algorithm Wenbo Xu, Jun Sun.....	435
Priority Assignment Strategy of Multiple Priority Queues Zhang Jianhua, Cong Yue.....	440
Design and Implement to Load Balancing to the Application Server Cluster Tang Wei.....	443
Finish Time Maximization Method: an Anti-Sequence Algorithm to Scheduling Task Graphs for Multiprocessors Jun Sun, Wenbo Xu, Bin Feng.....	447
Implementing and Invoking a Remote Object Calling Native Methods via RMI-IIOP and JNI Minglong Qi, Qingping Guo, Luo Zhong.....	451
A Technology to Improve Embedded-Linux Real-time Performance He Keyou, Hung Minfeng.....	457
7. Web-based Computing	
Exploring the Initial Structures of Dynamic Markov Modeling for Chinese Text Compression Ghim Hwee Ong, Junping Ng.....	460
Research on Integration of Web Services and Workflow Modeling Technique Shen Yuan, Chen Wenbo, Yao Zhiqiang.....	464

A Trust Management Framework Suitable for Web Services Security	
Ai Ping, Mao Yingchi.....	469
Research of Intelligent Search Engine Technology Facing Electronic Commerce	
Tong XiaoJun, Wang Zhu.....	474
Metadata Catalog Service for Geographic Information Resource	
Xu Kun, Liao Husheng, Du Jinlian.....	478
Research of Comparing CORBA with DCOM	
Wang Jingyang, Wang Xiaohong, Yuan Dun, et al.....	481
The Summarize of JAVA Platform for Web-based Computing	
Huiying Xu, Xinzhong Zhu.....	484
Semantic Web Enabled the Context Information in Ubiquitous Computing System	
Chen Xuhui, Tang Shancheng, Wang Yimin	488
A Web-based Engineering Optimization System and Its Application	
Caijun Xue, Hong Nie, Yanqin Dai.....	493
Research on the Model of Intelligent Meta-search Engine	
Li Liu, Wenbo Xu.....	497
A Study of CORBA Multi Port ORB Architecture Based on Hierarchy Domain	
Guo Yinzhang, Xie Liping, Xu Yubin, et al.....	501
A Multi-purpose Web Information Publishing Framework	
Yiqing Kong, Bin Feng, Wenbo Xu, et al.....	505
A Mediation-based Approach for Distributed Digital Library Services	
Yu Jianghong, Fang Wei.....	508
Web-Based Learning and Fault Diagnostic System	
Feng Pan, Wenbo Xu.....	511
A Design and Implementation of Dynamic E-business System Based on WEB Services	
Shaozhen Ye, Huajun Han.....	516
Research and Design for the Wrapper of Web-Based Data Resource Assembling Based on Soap	
ZhiHua Li, Jun Sun.....	521
An Approach towards Automated Web Services Composition	
Muhammad Adeel Talib, Yang Zongkai.....	524
Link-based Markov Model Prefetching Algorithm on Web Cache	
Wang Zhao, Guo Chengcheng, Yan Puliu.....	530
Analysis and Comparison between Two Distributed Object Technologies CORBA and DCOM	
He Keyou, Zhang Weilin.....	535

Volume II

8. Distributed Database.....	538
Deadlock Detection and Resolution in a Dike Safety Detection Management Information System Wu Jie, Liu Xiangsheng, Wu Wei.....	538
Mining Fuzzy Associate Rules for Anomaly Detection XiongPing, ZhuTianqing, HuangTianshu.....	541
A Pocket Spatial Database Prototype and its Query Language Guobao Yu, Husheng Liao, Yuming Zheng.....	545
Research and Implementation of Distributed Data Dissemination Jing Feng, Kong Yi, Chunhui Fan, et al.....	549
A Multi-granularity Locking Protocol Based on Ordered Sharing Locks in Engineering Databases that Supports Cooperative Design Chen Guoning, Li Taoshen, Liao Guoqiong.....	552
A Multi-dimension Perspective to XML Databases Modeling Liu Hongxing, Lu Yansheng.....	559
Construction and Maintenance of the Knowledge Base Used in GSIES-TOOL Xu Yong, Zhong Luo, Yang Ke.....	563
Create Distributed Application with Java RMI to Manipulate BLOBs Wang Jingyang, Wang Jianxia, Zhang Xiaoming, et al.....	567
Cooperation Agent Applications for MKA Based on Grid Computing Xia Huosong.....	570
Accessing BLOB Data Stored in Database Based on ADO in Visual C++ Qin Min, Wang Jingyang, Zhang Xiaoming, et al.....	574
Research on How to Connect Database in FORTRAN [*] Xia Hongxia, Jin Peng, Yuan JinLing.....	578
Methods of Applet Querying Database through Servlet Qin Min, Wang Jingyang, Wang Jianxia, et al.....	581
The Research of an Inventory Control Information System Based on the Internet Xiao Hanbin, Mo Lili, Zeng Xiangfeng.....	584
The Warehouse Management System Based on the Distributed Database Xiong Guohai, Wan Junli.....	588
The Design and Realization about the Campus Information Management System Based on the Data Warehouse Wang jianxia, Zhou wanzhen, Qin min, et al.....	591
Application of Distributed Database of Electric Power Management Information System Wu Wei, Wu Jie, Hu Peng.....	595
The Feature Parameter Extraction in Palm Shape Recognition System Wang Jianxia, Qin Min, Zhou Wanzhen, et al.....	598

9. E-Business	601
Commercial Bank Credit Risk Real-Time Value-at-Risk Computing System Rong Lan, Shouqi Zheng	601
Contract-based Interlayer: a Two-way Approach to Integrate Call Center with J2EE Framework Wu Cen, Lin Zuoquan, Zhao Xinyu, et al.	605
Design and Implementation of a General Secure Extensible Payment Gateway Architecture Bo Meng, Qianxing Xiong, Huanguo Zhang	610
The Study on Exchange Platform of Network Manufacturing Products Based on Digital Watermarking Techniques and Fair Exchange Protocol Zude Zhou, Zhiyang Wang, Quan Liu	614
Data Mining System Based on Web Services for E-commerce: Architectonics and Algorithm Luo Zhong, Qiwei Tong, Bin Fan, et al.	619
Building Robust J2EE Web Applications with Integration of Struts and JavaServer Faces Yang Hao, Guo Qingping	623
An XML Web Service Application Architecture Based on Microsoft BizTalk Server Zhou Ying, Liu Quan	627
A Model of the 3D Virtual Shopping that Has the Intelligent and Cooperative Purchasing Functionalities Zhao Yiming	630
A Server Electronic Wallet Architecture Supported Multi-payment Protocols and Instruments Bo Meng, Zhang Huanguo, Xiong Qianxing	634
ebXML: the Global Standard for Electronic Business Chen Caixian, Ran Chunyu	639
The Design and Implementation of Distributed Database in Trust Investment Synthesized Business System Kaiying Yang, Fahong Yu	643
10. E-Education	647
The Realization of CAI on Campus Net Wang Jing	647
Study on Focus in a SIP-based E-learning System Zeng Qingheng, Hu Ruimin	649
Java Based Distributed Learning Platform Zhang Xiaoming, Zhu Jinjun, Wang Jingyang, et al.	654
A Distributed Remote Education System Based on CSCW Rui Hao, Chunyu Ran, Qi Shen	658
Research and Design of Collaborative Learning System Kaiyan Wang, Guzi Huang	664
An Individual E-education System Based on Data Mining Zhang Xuemei, Zhang Xiaoming, Zhu Jinjun, et al.	667
The WSE and its Application on the Encryption in the Remote Education System Ran Chunyu, Bai Lin, Hao Rui	669

11. Distributed Applications in Engineering	672
Solution of the Wigner-Poisson Equations for RTDs M. S. Lasater, C. T. Kelley, A. G. Salinger, et al.....	672
One New Method and Its Parallelization of Perturbation Expansion for Coupled System of Acoustic and Structure Deng Li, Suzuki Masabumi, Hagiwara Ichiro.....	677
Parallel Reservoir Integrated Simulation Platform for One Million Grid Blocks Case Pan Feng, Cao Jianwen, Sun Jiachang.....	681
The Study of Distributed Hydrologic Data Integration Based on CORBA Lou Yuansheng, Wang Zhizian, Ai Ping, et al.....	685
Two Kinds of Novel Evolutionary Fuzzy Controllers — Control Algorithm Analysis Wang Pan, Xu Chengzhi, Zhang Jianjian, et al.....	690
A Study of the Mixed Fuzzy PID Controller in the Accurate Orientation Chen Yunji, Shen Keyu.....	694
Codesign for Complex Hard Real-time Embedded Systems Jin Yongxian.....	700
Development of a Distributed Embedded Remote Control Monitor System Based on CAN Bus Qin Juanying, Feng Xin, Wu Guoping.....	705
Study on Distributed Control System of Fuel Cell Electrical Vehicle Wu Youyu, Yang Jufang, Xie Changjun.....	710
Design of the Distributed Long Distance Water Supply Control System with Process Field Bus Technology Meng Hua, Yan Cuiying, Jia Hui ren, et al.....	715
The Research and Application of Real-time Monitor System Based on CAN Bus Network Tao Dexin, Cao Xiaohua, Mo Lili.....	719
Semi-active Logic Control Algorithm for MR Dampers Using Accelerations Feedback Chen Jing, Qu Weilian, Youlin.....	723
Estimate Model of Delay in Autolever System and its Algorithm Design Meigui Han, Jinglu Liu, Guangsheng Dongye.....	727
The Design of Power Battery Management System Based on Distributing CAN Bus Zeng Chunnian, Chen Yu, Qiao Guoyan.....	731
Application of the Distributed Parallel Processing in the DNA Sequence Alignment Mao Liming, Wang Zhongjun, Guo Qingping.....	735
Communication between WDPF and Remote Terminal Unit in Power Plat Yi Kui, Zhu Tianqing.....	739
The Health Monitoring and Damage Identification Platform of the Civil Infrastructure Based on Internet Xiao Chun, Qu Weilian, Zou Chengming.....	743
The Research on Intelligentized Distributed Cooperative Virtual Environment Gao Shu, Cheng Dingfang.....	746
Study on Application of Data Fusion in Erosion Detection of Furnace Lining Liu Quan, Zhang Xiaomei.....	749

Fiberglass Molding Technology and Control Method on Tank Kiln Chen Jing, Yuan Youxin.....	752
Design of a Distributed Monitoring and Control System Based on DSP Qin Juanying, Wu Guoping, Zhu Rongbo.....	755
Visual Federation Control Mechanism Feng Zhe, Xu Dongping.....	758
The Study and Application of OGSA Grid Based on Digital Manufacturing Zhang Fan, Zhou Zude, Liu Quan.....	761
12. Neural Network Computing.....	765
Research on NN and RKB Based Expert System of Resistance to Corrosion of Sulfate on Concrete Luo Zhong, Qiong Jiang, Fei Huang, et al.....	765
A New Neural Network Models Based for Rule-Based Reasoning Xiong Shanqing, Pan Hao, Zhang Yingjiang.....	768
Study the Application of Neural Network in the Prediction of Regional Integrated Transport Structure Huiyuan Jiang, Jiang Li.....	771
Using HTABP Algorithm to Determine Number of Hidden Units in NN Zhang Xi, Pan Hao, Xiong Shanqing.....	774
A Study on Neural Network Based on Contractive Mapping Genetic Algorithm Zou Chengming, Tong Qiwei, Yang Hongyun, et al.....	777
Study on Logistics System Safety Based on Neural Network and Fuzzy Probability Li Bo, Chen Dingfang, Zhang Xiaochuan, et al.....	780
Rsearch on a Back-Propagation Neural Network Based Q Learning Algorithm in Multi Agent System Lin Ouyang , Qingping Guo, Santai Ouyang.....	784
Recursive Neural Networks and its Application in Forecasting the State of Metal Oxide Arrester Zhou Long.....	790
13. Multi Agents.....	793
Multiagent-Based Partner Selection of Dynamic Alliances in Inter-organizational Collaborative E-commerce Wang Jie , Shi Xingguo, Zhong Weijun.....	793
Agile Reconstruction Methods Based on Agent in Distributed Database Qu Youtian, Xu Hong.....	798
The Applications of Multi-agent in an Expert System –AVDDT Yang Kaiying.....	802
Role-oriented Multi-agents Approach to Optimize for Grid Resource Allocation Qian Wang, Debao Xiao.....	806
Study on Multi-agent Based Environment for Long-distance Collaborative Learning on Internet Ruolin Ruan.....	810
A Heuristic Algorithm for Agent-based Task Scheduling in Grid Environments Ding Shunli, Yuan Jingbo, Ju Jiubin.....	814

14. Image Processing and Multimedia Applications.....	819
Parallel Hyperspectral Integrated Computational Imaging Bin Dai, Robert A. Lodder.....	819
A Binary Partitioning Approach to Image Compression Using Weighted Finite Automata for Large Images Ghim Hwee Ong, Kai Yang.....	824
A Fractal Rotating Vector Algorithm of Radar Echo Image Plotting Dan Liu, Dayong Zhang.....	828
An Iterated Algorithm for Implicit Surfaces Rendering Ni Tongguang, Gu Yaolin.....	831
An Implementation of Quarter Pixel Block Motion Estimation Using SIMD Xinchen Zhang, Ruimin Hu, Deren Li, et al.....	834
Rendering CG Objects into Photographs with Light Probe Images Wang Jun, Gu Yaolin.....	839
Network Architecture for Real-Time Distributed Visualization and 3D Rendering Lamei Yan, Xiaohong Zeng, M.Mat Deris.....	843
Image Segmentation Based on Fuzzy Maximum Entropy and Simulated Annealing Algorithm Liu Huikang, Li Juan, Wu Jin.....	846
The Study of Image Modeling in the Digital City Based on Distributed Computation Lu Feng, Ge Shun.....	850
Grey-Level Image Processing with a Parallel-Distributed System Model Ge Hongwei, Xu Wenbo.....	854
Approach of Feature Extracting Based on Single Sample Xu Dongping, Chen Jingliang.....	858
The Approach of Extracting the Graphs from Images of 3-D Objects Tao Hongjiu, Wen Yajuan, Tong Xiaojun.....	861
A Distributed Video Proxy System Based on Cache Jia Jiong, Zhu Jianxin.....	864
The Building of Inexpensive Large-scale Storage System for Video Applications Dong Xiaoming, Xie Changsheng.....	868
A Distributed Volume Visualization Architecture on the Grid Yaolin Gu, Zhe Cao.....	872
Analysis of Distributed Video-On-Demand System Based on Cluster Zheng Shijue, Ma Wei, Zhang Jiangling.....	876
The Investigation to Distributed Supervision System Based on GPRS Huie Chen, Congxin Liu, Weilu Zeng, et al.....	880
Distributed Virtual Reality Environments Based on VRML Fang Hua, Yaolin Gu.....	884
A Method of Computing Fractal Dimension Ren Wei, Liu Da, Xie Ling.....	888
Adaptive Partition and Hybrid Method in Fractal Video Compression Wang Meiqing, Liu Rong.....	891

An Improved Fractal Image Compression Approach by Using Iterated Function System and Genetic Algorithm Liu Guanrong, Zheng Yang, He Hua.....	897
Synchronization of Decompression and Display in Fractal Video Compression System Cheng Hang, Shu Zhibiao, Fang Yan, et al.....	903
The Cache-Multicast Method of Proxy Cache for Streaming Media Xu Zhiwen, Guo Xiaoxin, Pang Yunjie, et al.....	907
The Application of Distributed Streaming Media Technology in CAI Pan Wenhong, Zhang Jianhua.....	912
A Parallel Image Processing System Based on DSP Arrays Liu Zhi, Zhu Wei, Chen Shu.....	915
The Hardware Designing for Real Time FPGA Based Image Processing Tao Hongjiu, Bao Yuliang, Tong Xiaojun.....	917
The Ascertainment of Scale Sampling Step for Numerical Realization Adopting Binary Dot-and-Grid Sampling of the Continuous Wavelet Transform Pu Yifei, Yuan Xiao, Liao Ke, et al.....	921
Application of Parallel Distributed Technology in Simulation Engineering Wang Xuehui, Zhang Lei.....	926
Intelligent Grading Based on Image Recognition Zheng Guang, Kong Meijing, Fu Dong, et al.....	931
The Recognition and Decomposition of Mixed Pixels in the Remotely Sensed Images Based on Gray System Theory Gui Yufeng, Tao Jianfeng.....	934
A Distributed Tracking System for Indoor Augmented Reality Applications Jian Mao, Yaolin Gu.....	937
Approach on Visual Federation Member Relationship Xu Dongping, Qin Juan.....	943
15. Information and Network Security.....	947
Security Analysis and Improvement of Some Threshold Proxy Signature Schemes Xue Qingshui, Cao Zhenfu.....	947
Bilinear Pairings-based Threshold Proxy Signature Schemes with Known Signers Xue Qingshui, Cao Zhenfu, Qian Haifeng.....	953
A Load Balancing Algorithm for High Speed Intrusion Detection Gong Jian, Lu Sheng, Rui Suying.....	959
A Measure and Design Method of Security Protocol Wang Tao, Guo Heqing, Yao Songtao.....	964
Implement Distributed Parallel Computing Based on EJB Huang Zhehuang Jiang Xiufeng, Wang Meiqing.....	971
Stand Space Theory and its Application on SET Protocol Xu Feng, Li Dake, Huang Hao.....	976

Robust Hash Used In the Application of Digital Image Signature Wu Jin, Qiu Ya, Huang Honglin, et al.....	982
An Adaptive Digital Watermarking Algorithm Based on Wavelet Transform Guo Zhiqiang, Jiang Xuemei, Liu Quan	986
Research on an Agile Protocol for E-Commerce Security Wang Yong, Xiong Qianxing.....	990
Grid Security and Relevant Technology Tian Junfeng, Ma Yankun	994
Security Design Model of E-Government Collaborative Platform Su Jindian, Guo Heqing, Yu Shanshan.....	998
Study of Trust Model for Grid Security Zhu Dawei, Zhou Zude, Liu Quan.....	1002
Research on Secure Gateway Based on Real-time Embedded Systems Wu Yufeng, Wu Quan, Li Fangmin	1006
Data Encryption Algorithms for Internet-based Real-Time Systems Li Hongyan, Shuang H. Yang, Tan Liansheng	1010
Research on Grid Security for OGSA Fan Zuguang, Li Fangmin, Liu Quan	1015
Security Prototype Framework Design for Open Grid Services Architecture (OGSA) Yang Qiulin, Zhou Zude, Li Fangmin	1019
The Design and Implement of a Distance Education System Based on Improved MVC Pattern Ran Chunyu, Hu Hengying, Chen Caixian.....	1023
Application of Data Mining Technology to Intrusion Detection System Xia Hongxia, Shen Qi, Hao Rui	1027
Design and Implementation of CSP Module Rao Wenbi, Xiong Huiyue, Tang Chunming, et al.....	1031
A Robust Approach to Authentication of Binary Image for Multimedia Communication Wu Jin, Xia Beibei, Liu Jian, et al.....	1035
Researches on Parallel Intrusion Detection Methods Based-on Network Processor Li Fangmin, Zhang Huifu, Yang Ka.....	1040
A General Dynamic Secret Sharing Algorithm in Distributed System Li Xiaoxin, Guo Qingping, Zhang Feng.....	1045
Central 3A Platform Based-on SSO Zhang Yingjiang, Li Jun, Zheng Qiuhua, et al.....	1054
A New Security Solution of E-Commerce Based on Web Service Xiang Yang, Li Layuan, Hao Yulong, Shi Zhiqing.....	1060
Firewall System Based on IPv4/IPv6 Min Lianying, Chen Jiong.....	1063
16. Computation Theory.....	1066
Empirical Studies for Two Evolutionary Fuzzy Controllers Xu Huazhong, Wang Pan, Xu Chengzhi, et al.....	1066

The General M Set and Julia Sets Generated by Complex Iteration $Z_{N+1}=Z_N^{-2}+C$ Zhe Xu, X. G. Deng, .D. Liu, et al.....	1069
Generating Algorithm of IAGO Generating Space Xiao Xinping, Tang Weiqing.....	1074
Type Checking for Software System Specifications in Real-Time Process Algebra Chuanwen Liu, Xinming Tan.....	1077
A Definition and Study of a New Kind of Similarity Measure of Fuzzy Sets Tong Xiaojun, Gao Zunhai, Yuan Zhiyong.....	1084
The Grade Difference Format and MGM (1,n) Optimization Model Zhang Shemin, Tong Xiaojun	1087
An Object-oriented Knowledge Representation Method Xu Yong, Zhong Luo, Yang Ke.....	1092
Mutual Subset Hood of the Fuzzy Set Wei Yiliang.....	1096
Common Function Expression of Similarity Measure of Fuzzy Set at l^p - distance Wei Yiliang.....	1099
Equality of Vector and Similarity Measure of Fuzzy Sets Tong Xiaojun, Xu Xiaozeng, Li Zhijun.....	1102

PREFACE

High-performance computing is increasingly being used in all aspects of modern society. It is well known that the distributed parallel computing plays a main role in the HPC. In recent years, more and more attentions have been put on to the distributed parallel computing. I am confident that the distributed parallel computing will play an even greater role in the near future. Since distributed computing resources, once properly cooperated together, will achieve a great computing power and get a high ratio of performance/price in parallel computing. In fact the grid computing is a direct descendent of the distributed computing.

It is the second time for the DCABES international conference to be held in Wuhan China. We are gratified that this time nearly 400 papers submitted which cover a wide range of topics, such as Grid Computing, Mobile Computing, Parallel/Distributed Algorithms, Image Processing and Multimedia Applications, Parallel/Distributed Computational Methods in Engineering, System Architectures, Networking and Protocols, Web-Based Computing & E-Business, E-Education, Network Security and various types of applications etc.

All papers contained in this Proceedings are peer-reviewed and carefully chosen by members of Scientific Committee and external reviewers. Papers accepted or rejected are based on majority opinions of the referee's. All papers contained in this Proceedings give us a glimpse of what future technology and applications are being researched in the distributed parallel computing area in the world.

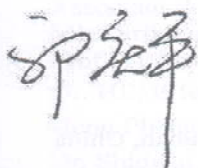
I would like to thank all members of the Scientific Committee, the local organizer committee, the external reviewers for selecting papers. Special thanks are due to Dr. Choi-Hong LAI, who co-chaired the Scientific Committee with me. It is indeed a pleasure to work with him and obtain his suggestions. Also sincere thanks should be forward to Tsui, Mr Y M Thomas, Chinese University of Hong Kong, for his enthusiastically taking part in and supporting the DCABES conference.

I am also grateful to Prof Peter Jimack, University of Leeds, UK; Prof Peter Kacsuk, LPDS, MTA SZTAKI, Hungary; Prof Simon Cox, University of Southampton, UK and Dr Simon See, Global Science and Technology Center, Sun Microsystems Inc, Singapore, for their contributions of keynote speeches in the conference.

Sincerely thanks should be forwarded to the Natural Science Foundation of China (NSFC), the China Ministry of Education (MOE), without their supports the DCABES 2004 could not be held in Wuhan China successfully. We would also like to thank the WUT (Wuhan University of Technology, China), the National Parallel Computing Society of China (NPCS), the ISTCA (International Science and Technology Cooperation of Hubei Province, China), and the CAA (Computer Academic Association of Hubei Province & Wuhan Metropolis, China) for their supports as local organizers of the conference. It should also be mentioned that the SUN Microsystem (Hong Kong Headquarter) made a contribution to the conference.

Finally I should also thank A/Professor Jian Guo for his efforts in conference organizing activities. The special thanks also should be given to my graduate students, Mr. Zhou Cheng for the conference website design, Mr. Zhou Cheng and Yang Hao for their efforts in organizing activities. It also should be mentioned that my graduate students, Mr. Ouyang Lin, Tang Guosheng, Yang Hao, Zhang Feng, Chen Jun, Shen Dingcai, Li Xiaoxin, Mao Liming, Zhang Feng and Ms Rao Jing of the grade 2002; Mr. Zhou Cheng, Wu Yanmao, Wang Qingsong, Liu Feng, Sun Hao, Cheng Haifeng, Han Guangming, Jiang Weijian Wu Weiwei of the grade 2003, and graduate student Guo Yucheng spent a lot of time and efforts typesetting the proceedings. Without their help the proceedings could not looks so good.

Enjoy your stay in Wuhan. Hope to meet you again at the DCABES 2005.



Guo, Professor Qingping
Chair of the DCABES2004
Dept. of Computer Science
Wuhan University of Technology
Wuhan, China

Honorary Chair

Zhou, Professor Zude, President of the WUT, China

Chair of Scientific Committee

Guo, Professor Q. P., Wuhan University of Technology

Co-Chair of Scientific Committee

Lai, Dr. Choi-Hong, University of Greenwich

Chair of Organizer Committee

Guo, Professor Q. P., Wuhan University of Technology

Scientific Committee (in alphabetical order)

Cai, Professor X.-C.	University of Colorado, Boulder, U.S.A.
Cao, Professor J.W.	Research and Development Centre for Parallel Algorithms and Software, Beijing, China
Chi, Professor X.B.	Academia Sinica, Beijing, China
Guo, Professor Q.P.	Wuhan University of Technology, Wuhan, China
Ho, Dr. P. T.	University of Hong Kong, Hong Kong, China
Kang, Professor L.S.	Wuhan University, Wuhan, China
Keyes, Professor D.E.	Columbia University, New York, USA
Lai, Dr. C.-H.	University of Greenwich, London, UK
Lee, Dr. John.	Hong Kong Polytechnic, Hong Kong, China
Liddell, Professor H. M.	Queen Mary College, University of London, London, UK
Lin, Dr. H.X.	Delft University of Technology, Delft, the Netherlands
Lin, Dr. P.	National University of Singapore, Singapore
Loo, Dr. Alfred	Hong Kong Lingnan University, Hong Kong, China
Ng, Dr. Michael	University of Hong Kong, Hong Kong, China
Sun, Professor J.	Academia Sinica, Beijing, China
Tsui, Thomas	Chinese University of Hong Kong, Hong Kong, China
Xu, Professor W.	Southern Yangtze University, Wuxi, China

Local Organizing Committee

Zhou, Professor Z.D. (Honorary Chair)	President of Wuhan University of Technology, Wuhan, China
Guo, Professor Q.P. (Chair)	Wuhan University of Technology, Wuhan, China
Zhong, Professor L. (Co-Chair)	Wuhan University of Technology, Wuhan, China
Liu, Professor Z.Y.	Wuhan University of Technology, Wuhan, China

Chen, Professor J.S.	Wuhan University of Technology, Wuhan, China
Kang, Professor L.S.	Wuhan University, Wuhan, China
Jin, Professor Hai	Hua Zhong University of Science and Technology, Wuhan, China
Liu, Professor Q.	Wuhan University of Technology, Wuhan, China
Lu, Professor J.G.	South Central China Nationality University, Wuhan, China
Tan, Professor L.S.	Central China Normal University, Wuhan, China
Xu, Prof. H.Z.	Wuhan University of Technology, Wuhan, China
Zeng, Professor C.N.	Wuhan University of Technology, Wuhan, China

REFEREES

Douglas, Professor Craig, Yale University, USA
Guo, Professor Q. P., Wuhan University of Technology, Wuhan, China
Ho, Dr. P. T., University of Hong Kong, Hong Kong, China
Jesshope Professor Chris R., Hull University, Hull, UK; Director of NZEdSoft, New Zealand
Kwan, Mr. W. K., University of Hong Kong, Hong Kong, China
Lai, Dr. Choi-Hong, University of Greenwich, London, UK
Lee, Dr. John, Hong Kong Polytechnic University, Hong Kong, China
Liddell, Professor Heather, Queen Mary and Westfield College, University of London, London, UK
Lin, Dr. Ping, National University of Singapore, Singapore
Loo, Dr. Alfred, Lingnan University, Hong Kong, China
Lu, Professor Zhengding, Huazhong University of Science and Technology, Wuhan, China
Ng, Dr. Michael, University of Hong Kong, Hong Kong, China
Paker, Professor Yakup, Computer Science Department, QMW, University of London, London, UK
Tan, Professor L.S., Central China Normal University, Wuhan, China
Jin, Professor Hai, Hua Zhong University of Science and Technology, Wuhan, China
Xu, Professor W., Southern Yangtze University, Wuxi, China
Ng, Dr. Michael, University of Hong Kong, Hong Kong, China
Cai, Professor X.-C., University of Colorado, Boulder, U.S.A
Cao, Professor J.W., Research and Development Centre for Parallel Algorithms and Software, Beijing, China
Keyes, Professor D.E., Columbia University, New York, USA
Lin, Dr. H.X., Delft University of Technology, Delft, Netherlands
Dr. Rüdiger Reischuk, Universität Lübeck, Germany
Prof. Dr. Joerg Rothe, Institut fuer Informatik, Universitaet Duesseldorf
A/Prof. Dr.Nayyer Masood, COMSATS Institute of Information Technology, Wah Cantt, Pakistan
Dr. Mike Brayshaw, QMW, University of London, London, UK
Dr. Ajay K Katangur, Department of Computer Science, Georgia State University, Atlanta, USA
Dr Bing Wang, Computer Science Department, University of Hull, Hull, UK
Associate Prof. Yuh-Shyan Chen, National Chung Cheng University Taiwan, China
Miss Srilaxmi Malladi, Georgia State University, Atlanta, USA
Prof. Yi Pan, Georgia State University, Atlanta, USA
Dr. HE, Lifeng, Faculty of Information and Computer Science, Aichi Prefectural University, Aichi, Japan
Heng Pheng Ann, Professor, the Chinese University of Hong Kong, Hong Kong, China
Dr. Shiduan Cheng, BUPT, Beijing, China
Dr Tang Ming Xi, School of Design, the Hong Kong Polytechnic University, Hong Kong, China
Dr. Christian Sohler, Computer Science Dept., UPB, Germany
Mr. Praveen Madiraju, Georgia State University, Atlanta, USA
Professor Madhusudhan Govindaraju, Binghamton, UK

Dr. Wong Tien Tsin, Chinese University of Hong Kong, Hong Kong, China



ISBN 7-5352-3269-8



9 787535 232694 >

ISBN 7-5352-3269-8/TP · 79
Price RMB 150/YUAN

P-GRADE: a High-level Grid Application Development Environment *

Peter Kacsuk

Lab of Parallel and Distributed Systems, MTA SZTAKI

Budapest, H-1111 Kende u 13, Hungary

Email: kacsuk@sztaki.hu Tel.: 36 1 329 7864

ABSTRACT

P-GRADE provides a high-level graphical environment to develop parallel applications transparently both for parallel systems and the Grid. One of the main advantages of P-GRADE is that the user does not have to learn the different APIs for parallel systems and the Grid. Simply by using the same environment will result in the generation of parallel applications transparently applicable either for supercomputers, clusters or the Grid. The P-GRADE portal enables the execution of parallel programs and workflows in several kinds of Grids including Condor Grids, Globus-2 and Globus-3 Grids as well as a Jini based Grid.

Keywords: Cluster and Grid programming, Grid portals, Grid execution.

1. INTRODUCTION

In the ten years history of Grid computing we could witness the rise and fall of three generations of Grid systems. First metacomputers, then resource-oriented Grid systems and recently service-oriented Grid systems appeared. This rapid change of various Grid systems makes the life of Grid users extremely difficult. Whenever a new Grid system has been deployed the users should learn new APIs and commands to develop and run their applications in the Grid. One remedy is the GAT (Grid Application Toolkit) API developed in the EU GridLab project [1]. The GAT will play a similar role in various Grids as MPI plays in various supercomputers. Another approach is the usage of a high-level Grid application development environment that can hide the details of various Grid systems.

Such a high-level Grid application development environment is P-GRADE (Parallel Grid Runtime and Application Development Environment) that helps the scientists, engineers or software developers who would like to quickly and conveniently develop a high-performance (parallel) application to run either on a supercomputer, cluster or Grid. More than that P-GRADE provides the necessary support to create scientific HPC workflows from existing PVM, MPI and sequential components.

Program developers can easily create by P-GRADE:

- parallel applications consisting of many processes or
- distributed workflow applications consisting of many components (parallel and sequential).

P-GRADE hides the low level details of how parallel/distributed systems are organized and managed so by

using P-GRADE end-users can easily access and submit jobs:

- to a supercomputer
- to a single cluster
- to a cluster of clusters
- to the Grid

P-GRADE also enables the exploitation of various Grids like Condor Grids [2] and Globus-2 Grids [3]. The P-GRADE portal that was specially designed for the Grid even gives access to service oriented Grids like GT-3 based Grids [4] and a Jini based Grid [5]. No matter which platform is selected to run the parallel application the end-users can observe her application at run-time:

- how processes interact:
 - within a parallel application
 - within a parallel workflow component
- how workflow components interact
- how the nodes of a cluster or sites of a Grid are utilized, etc.

The main functionalities of P-GRADE have been described in detail in [6] therefore in the current paper we concentrate on the configuration aspects of P-GRADE as well as its portal extension called as P-GRADE portal. Section 2 shortly summarizes the usage scenarios and execution modes of P-GRADE. Section 3 describes the typical configurations by which P-GRADE could advantageously be used. Section 4 introduces P-GRADE portal and its main functions.

2. P-GRADE USAGE SCENARIOS

P-GRADE is a high-level graphical programming environment that can be used in three different scenarios.

Scenario 1: The user can develop a high-performance (parallel) software component from scratch in her favourite language (FORTRAN, C, or C++) and use the high-level graphical notation of P-GRADE to express parallelism. User does not have to learn the low level details and concepts of PVM and MPI. By the P-GRADE code generator she can generate either PVM or MPI code from her graphical notation. Then the P-GRADE parallel debugger supports her to make her code bug-free. In the next stage of program development, the monitoring and visualization tools of P-GRADE enable the performance optimization of her code in a real parallel environment such as a cluster. Finally, the developed code can run under the P-GRADE supervision either on a supercomputer, cluster or in a Condor or Globus-2 Grid.

Scenario 2: The user can develop a high-performance (parallel) software application from an existing sequential code. In this case she does not have to rewrite everything from scratch. She can reuse the inherently sequential parts of the original code (no matter if it is in FORTRAN, C, or C++) and built into the parallel (graphical) framework that is provided by P-GRADE. In both Scenario 1 and 2 the user can access any existing library functions from her parallel program.

* The research described in this paper has been supported by the Hungarian Supercomputing Grid (OMFB-00728/2002) project, Hungarian IHM 4671/1/2003 project, and Hungarian OTKA T042459 project.

Scenario 3: The user can create a workflow from existing components and can run this workflow in a Condor or Globus-2 Grid. She can place sequential and parallel components into her workflow. The parallel components can be PVM or MPI programs, or components developed by P-GRADE. The latter is useful if there is a missing component in the workflow and the user wants to develop it in P-GRADE using the Scenario 1 or 2.

P-GRADE supports the **interactive** development of a parallel program as well as its job execution mode. The interactive execution can be on a single processor system, on a supercomputer or on a cluster. The recommendation is that the editing, compiling and debugging activities should be on a single processor desktop machine while mapping, and performance monitoring should take place on parallel systems like supercomputers or clusters. If the program is correct and delivers the expected performance on parallel systems, the user can switch to the **job mode**. Here the user should specify the resource requirements, input files, output files and error file of the job. Then P-GRADE automatically generates the appropriate job from the parallel program developed in the interactive working mode. The recommended applications of interactive and job mode are summarized in Figure 1. P-GRADE also supports the creation and execution of scientific workflows that are always executed in job mode and in Grid systems (see Figure 1).

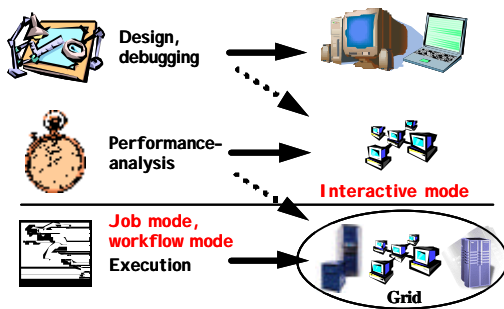


Figure 1. Usage of Interactive and Job Modes

P-GRADE programs can be executed on two types of resources:

- dedicated (interactive) resources
- Grid resources

Interactive resources are those that you can log into via `rsh` or `ssh` without a password (e.g. a remote cluster) or your local host that you are currently logged into. On dedicated (interactive) resources the user is able to develop and execute applications in *interactive* way.

If the user wants to access **Grid resources** (such as Condor or Globus resources) only the *job mode* of P-GRADE can be used. **Globus resources** are those you can access via Globus with your valid proxy certificate. This certificate must be stored and initialized on an interactive resource thus, a Globus resource can be configured if (and only if) the user already set up an interactive resource before accessing any Globus resource. The user can configure an interactive resource with Condor support as well, if it is a Condor submitter machine. In this case, the user will have access to a Condor-based cluster, i.e. a **Condor resource** via this interactive resource. Notice that an interactive resource is always needed even if the target is a Grid resource. The relationship of P-GRADE resources

are summarized in Figure 2.

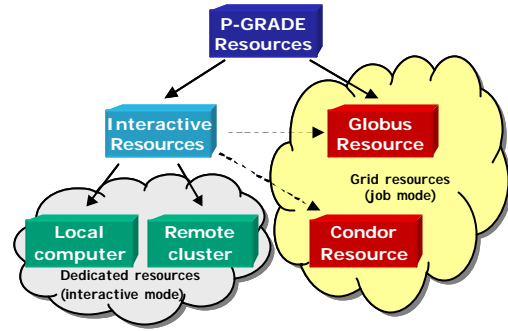


Figure 2. Relationship of P-GRADE Resources

3. TYPICAL P-GRADE CONFIGURATIONS

The various execution modes and P-GRADE resources can be used in 8 different typical configurations:

- Configuration 1: P-GRADE on a desktop
- Configuration 2: Dedicated cluster
- Configuration 3: Non-dedicated cluster
- Configuration 4: Clusters of Condor Grid
- Configuration 5: Sites of Globus-2 Grid
- Configuration 6: MPICH-G2
- Configuration 7: Workflow on Globus-2 Grid
- Configuration 8: P-GRADE portal

The various configurations are created according to the placement of the two main components of P-GRADE:

- P-GRADE GUI: graphical user interface
- P-GRADE RS: run-time system

P-GRADE GUI provides:

- GRED: a graphical environment to construct parallel programs and workflows
- a mapping tool GUI to assign processes of a parallel program to processors of a cluster or to assign components of a workflow to sites of a Grid
- DIWIDE: a debugger GUI to animate, watch and control the execution of parallel programs on a desktop or cluster
- PROVE: a visualization tool to visualize program execution

The P-GRADE RS provides the necessary support to run P-GRADE applications on a desktop machine, on a supercomputer, on a cluster or on a Grid. The run-time system includes

- GRP2C: a compiler to generate C code with PVM or MPI library calls
- A mapping facility to map processes or workflow components to resources
- DIWIDE: a distributed debugger to debug parallel and distributed applications
- GRM: a distributed monitoring infrastructure to collect run-time trace information on application execution both in desktop and dedicated cluster environments (In non-dedicated cluster or Grid environments a real Grid monitor support is needed. Such a Grid monitor is Mercury [7].)

- A checkpoint system to checkpoint PVM programs developed by P-GRADE
- A migration unit to automatically migrate processes of P-GRADE programs inside a cluster or among sites of a Grid
- A load-balancer unit to provide well balanced execution of the parallel program among the nodes of a cluster
- A workflow engine that controls the workflow execution and takes care of the necessary file transfers.

Configuration 1: P-GRADE on a desktop

In this configuration the user installs both the P-GRADE GUI and RS on the desktop. As a consequence the desktop can be used as a P-GRADE parallel program development environment. Every component of P-GRADE except for the workflow subsystem can be applied. The desktop configuration can be used to interactively create and debug parallel programs (first layer in Figure 1).

Configuration 2: Dedicated cluster

Dedicated cluster means that the whole cluster or several nodes of the cluster are dedicated to the execution of the parallel program. No other program can run simultaneously with this parallel program on the whole cluster or on the dedicated part of the cluster. In this configuration P-GRADE GUI is installed on the desktop and P-GRADE RS runs on the cluster.

The desktop is used to edit and map the application. The actual compilation and execution is performed in the dedicated cluster in interactive mode. The user can debug and monitor the program in the cluster environment. The P-GRADE load-balancer can be used in this configuration to provide well-balanced node exploitation. The P-GRADE checkpoint and migration unit can be used to migrate processes of PVM programs (developed by P-GRADE) among the nodes of the cluster.

This configuration enables:

- The performance testing of the parallel program (the performance analysis GUI runs on the desktop machine)
- The production execution of the parallel program on a dedicated cluster where no other program can run simultaneously with this parallel program (under such circumstances the use of P-GRADE load balancer can provide well balanced exploitation of the cluster nodes)

The parallel program monitoring is enabled by GRM and execution visualization is provided by PROVE. Even the migration and load-balancing activities can be visualized.

A variant of this configuration is the usage of an ultra-thin client on the desktop. In this case P-GRADE GUI and RS are installed on the dedicated cluster and the user can access them remotely via X redirection as shown in Figure 3.

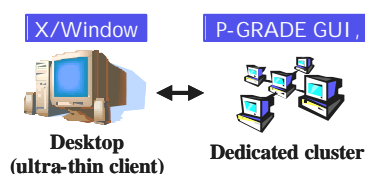


Figure 3 Dedicated Cluster with Ultra-thin Client

This configuration is recommended under certain conditions:

- There is reliable network connection with high bandwidth between the desktop and the dedicated cluster
- P-GRADE GUI cannot be executed on the desktop due to the specified software or performance requirements:
 - The operating system on the desktop is **not** Linux but the desktop has X/Window client installed.
 - The operating system on the desktop is old fashioned without the required libraries.
- The in-house security policy does not allow configure the cluster firewall as it required by P-GRADE.

Configuration 3: Non-dedicated cluster

Non-dedicated cluster means that several users' application jobs can simultaneously run on the cluster and typically a local job manager (such as Condor, SGE, PBS, etc.) takes care of queueing and launching user jobs. In this case P-GRADE GUI is installed on desktops from where users can simultaneously run their P-GRADE application on the cluster where P-GRADE RS is installed as shown in Figure 4.

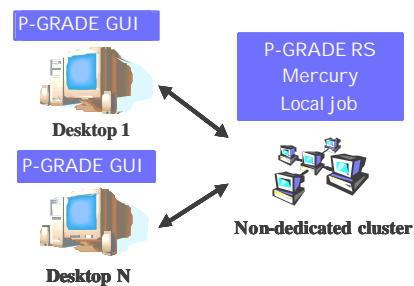


Figure 4. Non-dedicated Cluster Configuration

In this configuration the desktop is used to edit, compile, submit and visualize the application. The actual execution is performed on the non-dedicated cluster in job mode. The P-GRADE load-balancer is not used in this configuration since it is the task of the local job manager to take care of resource exploitation. The user can monitor the parallel program execution in the cluster environment if Mercury is installed there. In this case execution visualization is provided by PROVE. Even the migration activities can be visualized. The ultra-thin client version can be used here as well.

Configuration 4: Clusters of Condor Grid

Condor Grid is a cluster of clusters in which several non-dedicated Condor clusters (so-called Condor pools) are connected by Condor flocking. Condor flocking means that if a Condor pool is overloaded, jobs or processes of a job can be executed on another Condor pool. There is no firewall among the Condor pools.

In this configuration the desktop is used to edit, submit and visualize the application. The compilation and execution are performed on one or several of the connected Condor pools. The P-GRADE load-balancer is not used in this configuration since it is the task of the Condor job manager to take care of resource exploitation. The P-GRADE checkpoint and migration unit can be used to migrate PVM jobs (developed by P-GRADE) among Condor pools at any time during the

execution. The user can monitor the parallel program execution in the Condor Grid environment by Mercury and can visualize by PROVE. The distribution of P-GRADE GUI and RS is shown in Figure 5.

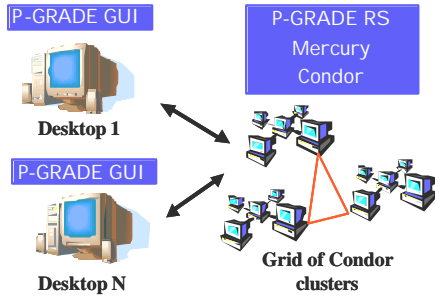


Figure 5. Configuration on Clusters of Condor Grid

Configuration 5: Sites of Globus-2 Grid

Globus-2 Grid is a heterogeneous Grid in which Grid sites (desktops, clusters, supercomputers, etc.) are connected by the Globus-2 Grid middleware. Typically the Globus sites are separated by firewalls and belong to various administrative domains.

In this configuration the desktop is used to edit, compile, submit and visualize the application. The parallel execution is performed on one selected site of the Globus Grid. The P-GRADE load-balancer is not used in this configuration since it is the task of the local job manager to take care of resource exploitation on the selected site. The P-GRADE checkpoint and migration unit can be used to migrate processes of a PVM job (developed by PGRADE) among nodes of a Grid site provided that the local job manager is Condor. The user can monitor the application execution in the Globus Grid environment by Mercury. The distribution of P-GRADE GUI and RS is shown in Figure 6. Notice that in this case the whole P-GRADE (GUI and RS) should be installed on the desktops together with the Globus client.

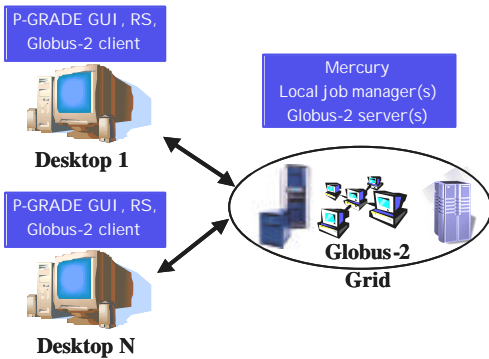


Figure 6. Configuration on Globus-2 Grid

Configuration 6: MPICH-G2

This configuration enables the production simultaneous execution of the parallel application on several Globus Grid sites by Globus-2 and MPICH-G2. The desktop is used to edit, compile, submit and visualize the application that is compiled into MPICH-G2 code. The parallel execution is performed simultaneously on several selected sites of the Globus Grid. The P-GRADE load-balancer, checkpoint and migration unit are not used in this configuration. The user can monitor the application execution in the Globus Grid environment by Mercury. The software configuration is very similar to

Configuration 5 but additionally MPICH-G2 is installed on the Grid sites.

Configuration 7: Workflow on Globus-2 Grid

This configuration enables the production execution of scientific workflows on several Globus Grid sites either sequentially or simultaneously depending on the job order defined by the workflow. The distribution of P-GRADE GUI and RS is shown in Figure 7. Notice that in this case the whole P-GRADE (GUI and RS) should be installed on the desktops together with the Globus client and Condor DAGMan and Condor-G. Condor DAGMan ensures that the execution order of jobs in the workflow will fit to the workflow dependency graph. Condor-G is used to submit the jobs of the workflow into the Globus-2 Grid.

In this configuration the desktop is used to create (edit, compile) and submit **workflows** and to visualize the workflow execution. Workflow (and component job) monitoring is performed by Mercury and execution visualization is enabled by PROVE.

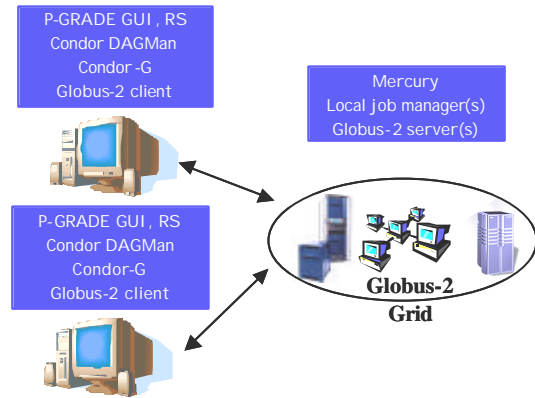


Figure 7. Workflow Configuration on Globus-2 Grid

4. P-GRADE PORTAL

Figure 7 shows that in case of Configuration 7 the user should install a very thick client. In order to avoid the need for such a thick client and to be able to use the PGRADE workflow facility from anywhere we developed the PGRADE portal. Figure 8 shows that in this case the thick client software is deployed on the portal server and the clients do not need anything else just a web browser.

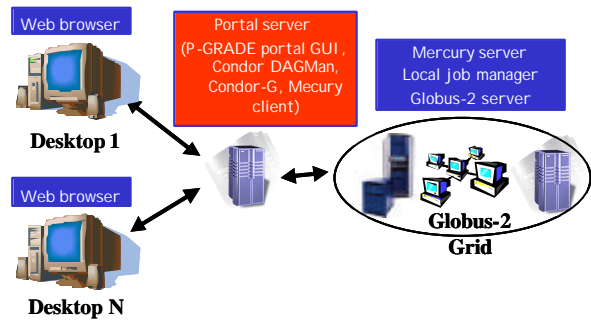


Figure 8. Portal Configuration

The P-GRADE portal is a workflow-oriented Grid portal where the main goal is to enable users to manage the whole lifecycle of creating and executing a complex application in

latter case when the job using the temporary file has been finished, the file is automatically removed from the Grid. After creating the workflow on the client machine it should be uploaded to the portal server machine as shown in Figure 9.

- Workflow management.** The uploaded workflows are managed by the Workflow Manager portlet that provides the following functionalities:

- Storing, updating and visualizing the status of the workflows and their component jobs.
- Submitting workflows when the user requests it (see Figure 9).
- Taking care of the necessary file transfers among the workflow jobs (executed on different Grid resources).
- Detach and attach running workflows.
- Delivering and storing the result files of the workflow execution.
- Showing workflow execution visualization and individual job execution visualization if the job is a parallel program.

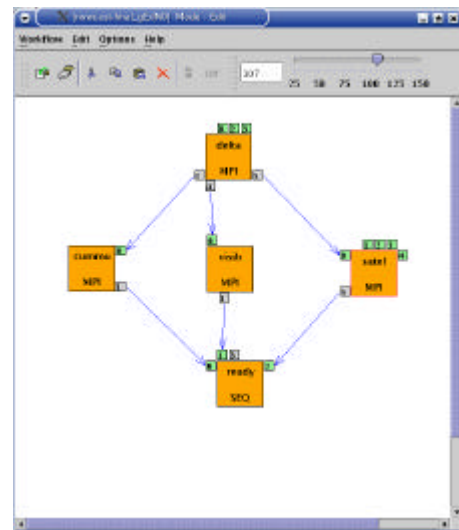


Figure 10. Workflow of a Meteorology Application

The Workflow Manager portlet collaborates with the Condor DAGMan [10] system that is responsible for selecting the next executable job of the workflow. Condor DAGMan provides a PRE-script facility to do job management before a job is actually started as a Condor-G job. This PRE-script can be written according to the actual needs in different Grid environments. If there is no Grid broker in the connected Grid, the user specifies at workflow edit time the Grid site where the job should be run. If there is a Grid broker, the Workflow Manager portlet uses the PRE-script facility to contact the Grid broker and asks the selection of an appropriate Grid site. Once the Grid site is selected (in either way) the Workflow Manager portlet uses again the PRE-script facility to transfer the necessary input files by GridFTP into the selected Grid resource. Then the job is submitted as a Condor-G job to the Globus-2 gatekeeper of the selected Grid site.

This is the way how the portal Workflow Manager portlet collaborates with a GT-2 Grid environment. The P-GRADE portal was actually connected so far to three different GT-2 based Grid systems: Hungarian Supercomputing Grid, LCG-2, and the EU GridLab testbed.

Workflow editing. The workflows can be graphically created at the client machine by the Workflow Editor written as a Java Web-Start application. A simple workflow used in a real-life meteorology application [9] is shown in Figure 10. Large squares represent jobs and small squares represent the input and output files. Definition of a job requires the identification of the job (e.g. delta), the location of the executable code and its type (SEQ, MPI, etc.). If there is no broker in the connected Grid system, the user should explicitly define the Grid resource where the job is to be executed. Files are identified by their logical name and location. It should be also defined if the file is a permanent one or used only temporarily. In the

5. CONCLUSIONS

P-GRADE provides a high-level graphical environment to develop parallel applications transparently both for parallel systems and the Grid. One of the main advantages of P-GRADE is that the user does not have to learn the different APIs for parallel systems and the Grid. Simply by using the same environment will result in the generation of parallel applications transparently applicable either for supercomputers, clusters or the Grid. The current version of P-GRADE supports the interactive execution of parallel programs both on supercomputers, clusters and Globus-2 Grids as well as the creation of a Condor or Condor-G job to execute parallel programs as jobs in various (Globus-2 and Condor) Grids.

P-GRADE is currently ported to the Hungarian ClusterGrid that connects the Condor pools of the Hungarian higher educational institutions into a high-performance, high-throughput Grid system. P-GRADE is already tested in the Hungarian Supercomputing Grid that combines the Hungarian supercomputers and large clusters into a Globus-2 Grid.

P-GRADE has been applied for the parallelisation of the MEANDER nowcast program package of the Hungarian Meteorology Service [9] and for the parallelisation of the Madcity urban traffic simulation system [11]. Recently several molecule dynamics applications and nuclear technology simulation code have been developed by P-GRADE at several Hungarian institutes. P-GRADE is intensively used by several UK, Hungarian and Polish universities for teaching parallel, distributed and Grid programming.

P-GRADE can be downloaded (with a User's Manual, Installation Guide and a set of demo programs) from the <http://www.lpds.sztaki.hu/pgrade/index.php> web site.

The P-GRADE portal enables the portability of P-GRADE workflows among various kind of Grids. It was connected not only to Globus-2 Grids but also to an OGSA compliant Globus-3 Grid in the framework of the UK e-science OGSA testbed project. The connection of the P-GRADE portal to the Condor pool based Hungarian ClusterGrid as well as to the service-oriented Hungarian JGrid (a Jini based Grid [12]) is under development. The ultimate goal is that a user should be able to access several different Grids simultaneously by the P-GRADE portal.

6. REFERENCES

- [1] G. Allen, et al, GridLab – “A Grid Application Toolkit and Testbed”, accepted for Special Issue on Grid Computing “Future Generation Computing Systems”
- [2] T. Maray et al, “Achievements of the Hungarian ClusterGrid Infrastructure Project”, Proc. of MIPRO'2004 Hypermedia and Grid Systems, Opatija, 2004, pp.208-212.
- [3] J. Patvarczki et al, “The Hungarian Supercomputing Grid in the Actual Practice”, Proc. of MIPRO'2004 Hypermedia and Grid Systems, Opatija, 2004, pp.203-207.
- [4] T. Delaitre, et al, “GEMICA: Grid Execution Management for Legacy Code Architecture Design, Conf. Proc. of the 30th EUROMICRO Conference, 2004, Rennes, France.
- [5] G. Sipos and P. Kacsuk, “Using Jini to Connect Condor Pools into a Computational Grid”, Proc. of MIPRO'2004 Hypermedia and Grid Systems, Opatija, 2004, pp.197-202.
- [6] P. Kacsuk et al, “P-GRADE: A Grid Programming Environment”, Journal of Grid Computing, Vol.1, No.2, 2003, pp.171-197.
- [7] Z. Balaton and G. Gombás, “Resource and Job Monitoring in the Grid”, Proc. of EuroPar'2003, Klagenfurt, 2003, pp. 404-411.
- [8] <http://www.gridisphere.org/gridsphere>
- [9] R. Lovas, et al., “Application of P-GRADE Development Environment in Meteorology”, Proc. of DAPSYS'2002, Linz, 2002, pp. 30-37.
- [10] www.cs.wisc.edu/condor/manual
- [11] T. Delaitre, et al, “Traffic Simulation in P-Grade as a Grid Service”, Conf. Proc. of the DAPSYS 2004 Conference, Budapest, September 19-22, 2004,
- [12] <http://jgrid.jini.org/>



Prof. Dr. Peter KACSUK is the Head of the Lab. of the Parallel and Distributed Systems in MTA SZTAKI. He received his MSc and university doctorate degrees from the Technical University of Budapest in 1976 and 1984, respectively. He received the kandidat degree and the Doctor of Academy degree (DSc) from the Hungarian Academy in 1989 and 2001, respectively. He habilitated at the University of Vienna in 1997. He is a part-time full professor at the University of Miskolc (Hungary), at the Eötvös Lóránd Univ. of Science in Budapest (Hungary) and at the Cavendish School of Computer Science of the University of Westminster (UK). He has been published two books and more than 180 scientific papers. He is the founder and director of the Hungarian Grid Competence Centre and co-editor-in-chief of the Journal of Grid Computing.

High Performance and Grid Computing Where Are We Going?

Dr Simon See

Global Science and Technology Center, Sun Microsystems Inc, Singapore

Email: simon.see@sun.com Tel.: +65-2397886

ABSTRACT

Over the last decades, high performance computing and internet has transformed the way science research, engineering and business is being conducted. We have seen the tremendous growth of computer technology both in hardware and software to address our needs. In this paper, the author attempted to walk through some of the work

Keywords: Grid Computing, Globus

1. MICROPROCESSOR- THROUGHPUT WAY

More than three decades ago, Moore's Law pronounced that the number of transistors that will fit per square inch on an integrated circuit will double about every two years. Since then, chip manufacturers have entered into a megahertz race, exploiting Moore's Law to deliver processors that now run at gigahertz speeds.. In addition to increasing processor frequency, microprocessor designer has wisely used increasing transistor gains to enhance scalability, memory management, and reliability for the vendors (e.g UltraSPARC) processors and, consequently, its systems. As a result, solutions deliver excellent application performance, even outperforming competing systems running at higher frequencies. At the same time, however, like other processor design teams, engineers have encountered the frustrating trend of diminishing returns. While Moore's Law effectively enables chip speeds to double about every two years, actual system performance gains are much lower due to significantly slower increases in memory speeds. In fact, memory speeds have only been doubling every six years—one-third the rate of microprocessors.

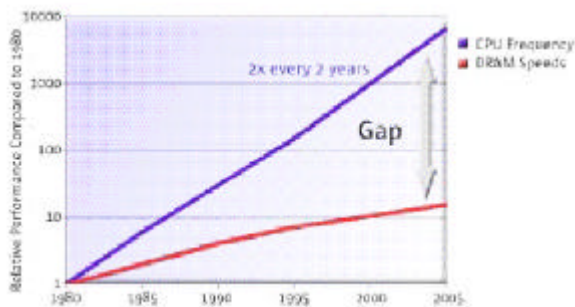


Figure 1 Gap of CPU and Memory

This gap in processor and memory speeds leaves ultrafast processors spending as much as 75 percent of the time waiting for memory to return required data instead of doing useful work. The chart below illustrates today's diminishing returns, with a 2.2-GHz Pentium 4 processor offering an 83 percent increase in frequency over a 1.2-GHz Celeron processor but only delivering a 20 percent gain in actual application performance. Even worse, this increase comes at a price spike of 446 percent and a power consumption hike of 84 percent.

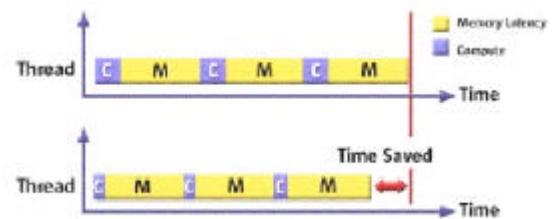
Processor	Celeron	Pentium 4	% Difference
Frequency	1.2 GHz	2.2 GHz	83%
Performance (Business Winstone 2001*)	44.7	54.8	20%
# of Transistors	41M	42M	5%
Power	29.9W	55.1W	84%
Cost**	\$103	\$562	446%

*Source: PC Magazine (February 2002)

Business Winstone 2001 measures system performance on several office-based applications, including Lotus Notes, Microsoft Office, and Norton AntiVirus.

**Cost Source: CNET (January 4, 2002); Yahoo Finance News (Jan. 8, 2002)

Figure 2 Power and performance



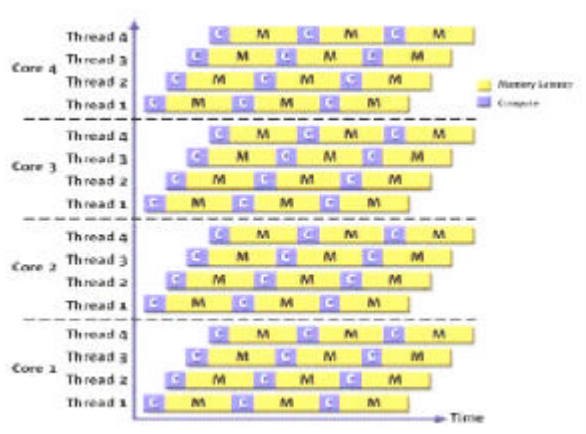
Thus the multithreaded processor has recently gain tremendous interest. A multithreaded processor is able to concurrently execute instructions of different threads of control within a single pipeline. The minimal requirement for such a processor is the ability to pursue two or more threads of control in parallel within the processor pipeline, that is, the processor must provide two or more independent program counters, an internal tagging mechanism to distinguish instructions of different threads within the pipeline, and a mechanism that triggers a thread switch. Thread-switch overhead must be very low, from zero to only a few cycles. Multithreaded processor features often, but not always, multiple register sets on the processor chip.

The current interest in hardware multithreading —Latency reduction is an important task when designing a microprocessor. Latencies arise from data dependencies between instructions within a single thread of control. Long latencies are caused by memory accesses that miss in the cache and by long running instructions. Short latencies may be bridged within a superscalar processor by executing succeeding, nondependent instructions of the same thread. Long latencies, however, stall the processor and lessen its performance. —Shared-memory multiprocessors suffer from memory access latencies that are several times longer than in a single-processor system. When accessing a nonlocal memory module in a distributed-shared memory system, the memory latency is enhanced by the transfer time through the communication network. Additional latencies arise in a shared memory multiprocessor from thread synchronizations, which cause idle times for the waiting thread. One solution to fill

these idle times is to switch to another thread. However, a thread switch on a conventional processor causes saving of all registers, loading the new register values, and several more administrative tasks that often require too much time to prove this method as an efficient solution.

Recently Sun has been designing processors built specifically to maximize throughput for network computing workloads. These chip multithreading (CMT) processors will leverage the increasing number of transistors supported on a microchip to process tens of threads simultaneously. This support for multithreading at the chip level is designed to dramatically increase application throughput, delivering revolutionary benefits to customers instead of incremental increases in speed. Take, for example, a standard Web-based business that runs multiple servers to handle customer requests. Each time a customer logs on to the site, a thread is sent to a processor to serve up Web page data. With 100 of today's single-processor servers, the company can handle 100 different threads at a time. When those traditional processors are replaced with CMT processors, each Web server will instead be able to execute tens of software threads simultaneously, significantly boosting performance while supporting server consolidation.

Sun's CMT processors will also have multiple cores on a single piece of silicon, with each core being able to process multiple threads. As a result, a single CMT processor will be able to process tens of threads simultaneously, exponentially increasing the amount of data processed each second.



2. GRID

The Internet and the World Wide Web have improved dramatically over the last few years, mainly because of increasing network bandwidth, powerful computers, software, and user acceptance. These elements are currently converging and enabling a new global infrastructure called "The Grid", originally derived from the electrical "Power Grid" which provides electricity to every wall socket. A computational or data grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to computational capabilities, as described in Foster and Kesselman [1]. It connects distributed computers, storage devices, mobile devices, instruments, sensors, databases, and software applications.

We have learned that grids provide many more benefits than

just the increase in resource utilization, or using idle resources, as described in many press articles today. Some of the key advantages a grid can provide are:

- Access: Seamless, transparent, remote, secure, wireless access to computing, data, experiments, instruments, sensors, etc.
- Virtualization: Access to compute and data services, not the servers themselves, without caring about the infrastructure.
- On Demand: Get resources you need, when you need them, at the quality you need.
- Sharing: Enable collaboration of (virtual) teams, over the Internet, to jointly work on one complex task.
- Failover: In case of system failure, migrate and restart applications automatically, on another system.
- Heterogeneity: In large and complex grids, resources are heterogeneous (platforms, operating systems, devices, software, etc.). Users can choose the best suited system for their specific application.
- Utilization: Grids are known to increase average utilization from some 20% towards 80% and more. For example, our internal Sun Enterprise Grid (with currently more than 7,000 processors in three different locations) to design Sun's next-generation processors is utilized at over 95%, on average.

These benefits translate into high-level value propositions which are especially beneficial to upper management in research and industry who has to make the decision to adopt and implement a grid architecture within the enterprise. Such values are:

- Increase agility (shorter time to market, improve quality and innovation, reduce cost, increase Return On Investment, reduce Total Cost of Ownership)
- Reduce risk (better business decisions, faster than competition) Enable innovation (develop new capabilities, do things previously not possible)..

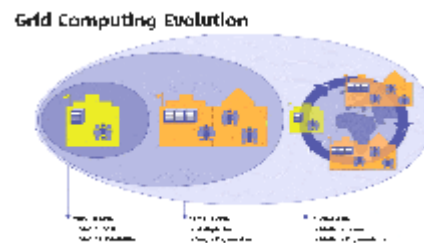


Figure 3 Evolution of Grid

Several of these benefits are present already in small managed compute cluster environments, often called Mini-Grids, or Cluster Grids, or Department Grids. In many of our early grid projects, since about 1998, our partners and customers started building mini-grids, see [2]. In fact, today (May 2003), over 7,000 cluster grids are in production, running the Distributed Resource Management (DRM) software Sun Grid Engine [3], or its open source version Grid Engine [4]. A few hundred of those early adopters already implemented the next level, so-called Campus or Enterprise Grids, connecting resources distributed over the university campus or the global enterprise, using the Sun Grid Engine Enterprise Edition [3]. And a few dozen of them are currently transitioning towards Global Grids, connecting resources distributed beyond university or enterprise firewalls, and using global grid technology like

Globus [5] and Avaki [6], integrated with Sun Grid Engine. This strategy of evolutionary transition from cluster, to enterprise, to global grids is summarized in Figure 3

A Global Grid Architecture

What are the motivations for creating a grid? What are the major areas of difficulty which must be addressed? What are the currently available components of a global compute grid and how do they address major user requirements?

The Globus Project [5] introduces the term Virtual Organisation (VO) as a set of users in multiple network domains who wish to share some of their resources. The virtual organisation may be (as is currently the norm) a group of academic institutions who wish to enable resource sharing. The possible functional objectives of such resource sharing include:

- The aggregation of compute power. This can result in a number of benefits:
 - Increase the throughput of users' jobs by maximizing resource utilisation
 - Increase the range of complementary hardware available e.g. compute clusters, large shared memory servers, parallel computers, etc.
 - Provide a (virtual) supercomputer grid which presents a platform for grand challenge applications
- The tight integration of geographically and functionally disparate databases
- The catering for a huge, dynamic dataset and the processing thereof.

The main software components (software tools) of a global compute grid are:

- User Interface: Enabling remote, transparent, secure access to the grid resources by non-expert users, through some straightforward interface, usually a web-portal.
- Broker: Automating job scheduling based upon the users' policies. Such policies could describe the users' priorities in terms of job requirements, available budget, time requirements, applications, etc. The broker would use these policies when negotiating on the users' behalf for a resource on the grid.
- Security, data-management, job-management and resource discovery. These are the key issues that have been addressed by the Globus project.
- Resource guarantees and accounting. This is an area of current research activity and links in with the brokering technologies.

Core Components for Building a Grid

To provide grid functionalities and benefits described in the previous section, a set of core middleware components is necessary. In our grid projects, we are using, among others, the following components:

- Access Portal: Grid Engine Portal, a few thousand lines of Java code for the Graphical User Interface, available in open source, [4], with the functionality to plug into any Portal Server, e.g. Sun ONE or Apache Portal Server, for additional security, authentication, authorization, and more.
- Globus Toolkit: With the Globus Security Infrastructure (GSI), the Globus Resource Allocation Manager (GRAM), the Monitoring and Discovery Services (MDS), and GridFTP for enabling efficient file transfer.
- Distributed Resource Management: E.g. Sun Grid Engine,

Sun Grid Engine Enterprise Edition, Condor [7], LSF [8], or PBS [9].

Distributed Resource Managers

The core of any Cluster or Enterprise Grid is the Distributed Resource Manager (DRM). Examples of DRMs are Sun Grid Engine [3], Platform Computing's Load Sharing Facility [8], or Altair's PBS Pro [9]. In a global compute grid it is often beneficial to take advantage of the features provided by the local DRMs. Such features may include the ability to

- create user sets whose access rights to the Cluster Grid may be controlled - this will strongly complement the limited authorisation currently available through Globus.
- provide resource guarantees to grid users.
- reserve portions of the local resource for local users.
- perform advanced reservation of compute resources.

One of the key advantages of local DRM software is that it can simplify the implementation of the Globus layer above it. Specifically, where the underlying compute resources are heterogeneous in terms of operating platform, processor architecture, and memory, the local DRM provides a virtualisation of these resources, usually by means of the queue concept.

Different DRMs have different definitions of a queue, but essentially a queue, and its associated attributes, represents the underlying compute resource to which jobs are submitted. If a Virtual Organization (VO) chooses to implement a specific DRM at each of its Cluster Grids, then the concept of implementing a virtualization of all the Cluster Grids is relatively straight forward despite the possibility that the underlying hardware may be quite heterogeneous. One simply aggregates all the queue information across the VO. Since the attributes of the queues will have a common definition across the VO, the interface to this grid could be designed to be analogous to that implemented at the campus level

As an example of a DRM, Sun N1 Grid Engine is a distributed resource management software, which recognizes resource requests, and maps compute jobs to the least-loaded and best suited system in the network. Queuing, scheduling, and prioritizing modules help to provide easy access, increase utilization, and virtualize the underlying resources. Sun Grid Engine Enterprise Edition, in addition, provides a Policy Management module for equitable, enforceable sharing of resources among groups and projects, aligns resources with corporate business goals, via policies, and supports resource planning and accounting.

There are mainly two solutions to integrate the local DRM with Globus:

- (a) There is an integration of the DRM with GRAM. This means that jobs submitted to Globus (using the Globus Resource Specification Language, RSL) can be passed on to the DRM. Evidently the key here is to provide a means of translation between RSL and the language understood by the DRM. These are implemented in Globus using GRAM Job manager scripts.
- (b) There is an integration with MDS. The use of a GRAM Reporter allows information about a DRM to be gathered and published in the MDS. The reporter will run at each campus site periodically via cron, and query the local DRM. This means that up-to-date queue information can be gathered across

many Cluster Grids.

Portal Software and Authentication

The portal solution may be split into two parts. Firstly, the web-server and/or container which serves the pages. Examples include Sun ONE Portal Server, Tomcat/Apache, uPortal. Second, the collection of Java servlets, web-services components, java beans etc. that make up the interface between the user and the Globus Toolkit and runs within the Server. The Grid Portal Development Kit is an example of a portal implementation which is interfaced with the Globus Toolkit.

2.1.3 The Globus Toolkit 2.0

In our early grid installations, we used Globus Toolkit 2.0 and 2.2 (GT2.0, GT2.2). The Globus Toolkit is an open architecture, open source software toolkit developed by the Globus Project. A brief explanation of GT2.0 is given here for completeness. Full description of the Globus Toolkit can be found at the Globus web site [5]. The next generation of the Globus Toolkit, GT3.0 GT3.0 re-implements much of the functionality of GT2.x but is based upon the Open Grid Services Architecture, OGSA, [10].

White Rose Grid

The White Rose Grid (WRG, [45]), based in Yorkshire, UK is a virtual organisation comprising of three Universities: The Universities of Leeds, York and Sheffield. There are four significant compute resources (Cluster Grids) each named after a white rose. Two cluster grids are sited at Leeds (Maxima and Snowdon) and one each at York (Pascali) and Sheffield (Titania).

The White Rose Grid is heterogeneous in terms of underlying hardware and operating platform. Whilst Maxima, Pascali and Titania are built from a combination of large symmetric memory Sun servers and storage/backup, Snowdon comprises a Linux/Intel based compute cluster interconnected with Myricom Myrinet.

The software architecture can be viewed as four independent Cluster Grids interconnected through global grid middleware and accessible, optionally through a portal interface. All the grid middleware implemented at White Rose is available in open source form.

The WRG software stack, Figure 4, is composed largely of open source software. To provide a stable HPC platform for local users at each site Grid Engine Enterprise Edition [3], [4], HPC ClusterTool [30] and SunONE Studio [46] provide DRM and MPI support and compile/debug capabilities.

Users at each Campus use the Grid Engine interface (command line or GUI) to access their local resource. White Rose Grid users have the option of accessing the facility via the portal. The Portal interface to the White Rose Grid has been created using the Grid Portal Development Kit [27] (GPDK) running on Apache Tomcat [47]. GPDK has been updated to work with Globus Toolkit 2.0 and also modified to integrate with various e-science applications.

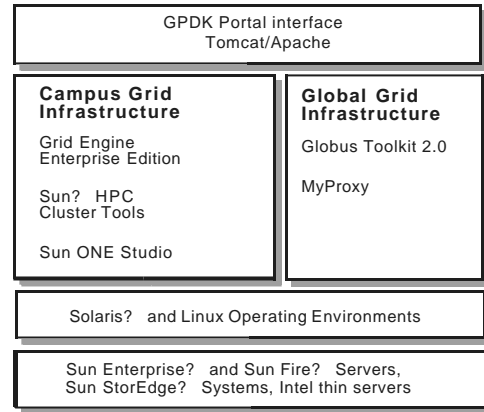


Figure 4 White Rose Grid Components, Hardware/software Stack.

Each of the four WRG cluster grids has an installation of Grid Engine Enterprise Edition. Globus Toolkit 2.0 provides the means to securely access each of the Cluster Grids through the portal.

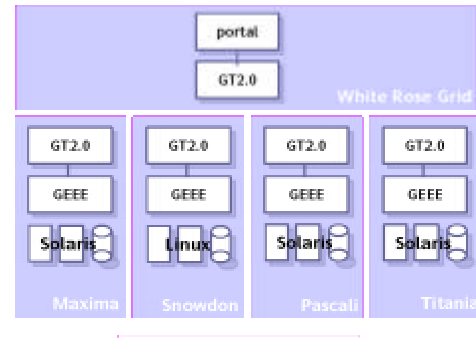


Figure 5 The Four Cluster Grid Computing Nodes of the White Rose Grid.

N1 Grid Engine

N1 Grid Engine is installed at each of the four nodes, Maxima, Snowdon, Titania, Pascali. The command line and GUI of Enterprise Edition is the main access point to each node for local users. The Enterprise Edition version of Grid Engine provides policy driven resource management at the node level. There are four policy types which may be implemented:

- **Share Tree Policy:** Enterprise Edition keeps track of how much usage users/projects have already received. At each scheduling interval, the Scheduler adjusts all jobs' share of resources to ensure that users/groups and projects get very close to their allocated share of the system over the accumulation period.
- **Functional Policy:** Functional scheduling, sometimes called priority scheduling, is a non-feedback scheme (i.e. No account taken of past usage) for determining a job's importance by its association with the submitting user/project/department.
- **Deadline Policy:** Deadline scheduling ensures that a job is completed by a certain time by starting it soon enough and giving it enough resources to finish on time. Override policy:
- **Override Policy:** Override scheduling allows the

Enterprise Edition operator to dynamically adjust the relative importance of an individual job or of all the jobs associated with a user/department/project.

At White Rose, the Share Tree policy is used to manage the resource share allocation at each node, Figure 5. Users across the three Universities are of two types: (a) local users are those users who have access only to the local facility (b) WRG users are users who are allowed access to any node in the WRG. Each WRG node administrator has allocated 25% of their node's compute resource for WRG users. The remaining 75% share can be allocated as required across the local academic groups and departments. The WRG administrators also agree upon the half-life associated with SGEEE so that past usage of the resources is taken into account consistently across the WRG.

Globus

As depicted in Figure 5, each WRG Cluster Grid hosts a Globus Gatekeeper. The default job-manager for each of these gatekeepers is set to Grid Engine using the existing scripts in the GT2.0 distribution. In order that the Globus jobmanager is able to submit jobs to the local DRM, it is simply necessary to ensure that the Globus gatekeeper server is registered as a submit host at the local Grid Engine master node. The Globus grid-security file referenced by the gatekeeper servers includes the names of all WRG users. New users' grid identities must be distributed across the grid in order for them to be successfully authenticated. Additionally to this, at each site all WRG users are added to the userset associated with the WRG share of the Enterprise Edition controlled resource. This ensures that the sum usage by WRG users at any cluster grid does not exceed 25%.

Portal interface

The portal technology used at White Rose has been implemented using the Grid Portal Development Kit. GPDK has been designed as a web interface to Globus. GPDK uses Java Server Pages (JSP) and Java Beans and runs in Apache Tomcat, the open source web application server developed by Sun Microsystems. GPDK takes full advantage of the Java implementation of the Globus CoG toolkit.

GPDK Java Beans are responsible for the functionality of the portal and can be grouped into the five categories; Security, User Profiles, Job Submission, File Transfer, and Information Services. For security, GPDK integrates with MyProxy, [48]. MyProxy enables the Portal server to interact with the MyProxy server to obtain delegated credentials in order to authenticate on the user's behalf.

Some development work has been done in order to port the publicly available GPDK to GT2.0. Specifically:

- GPDK was modified to work with the updated MDS in GT2.0
- Information Providers were written to enable Grid Engine Queue information to be passed to MDS. Grid users can query MDS to establish the state of the DRMs at each ClusterGrid.

As with many current Portal projects, the WRG uses the MyProxy Toolkit as the basis for security. Prior to interacting with the WRG, a user must first securely pass a delegated credential to the portal server so that the portal can act upon that user's behalf subsequently. The MyProxy Toolkit enables this.

The event sequence up to job submission is as follows:

1. When the user initially logs on, the MyProxy Toolkit is invoked so that the portal server can securely access a proxy credential for that user.
2. The users can view the available resources and their dynamic properties via the portal. The Globus MDS pillar provides the GIIS, LDAP based hierarchical database which must be queried by the portal server.
3. Once the user has determined the preferred resource, the job can be submitted. The job information is passed down to the selected cluster grid where the local Globus gatekeeper authenticates the users and passes the job information to Grid Engine Enterprise Edition.

3. REFERENCES

- [1] Ian Foster, Carl Kesselman, "The GRID: Blueprint for a new Computing Infrastructure," Morgan Kaufman Publishers, 1999.
- [2] Customer grid examples, <http://www.sun.com/software/grid/success.html>
- [3] Sun Grid Engine website, <http://www.sun.com/grid>
- [4] Grid Engine open source project at <http://gridengine.sunsource.net/>
- [5] Globus website, <http://www.globus.org>
- [6] Avaki website, <http://www.avaki.com>
- [7] Condor website, <http://www.cs.wisc.edu/condor/>
- [8] LSF website, <http://www.platform.com/products/wm/LSF/index.asp>
- [9] PBS website, <http://www.altair.com/pbspro.htm>
- [10] OGSA website, <http://www.globus.org/ogsa/>
- [11] GlobeXplorer website, <http://www.globexplorer.com>
- [12] OpenGIS website, <http://www.opengis.org>
- [13] OpenLDAP website, <http://www.openldap.org>
- [14] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," International Journal of Supercomputer Applications, 15(3), 2001.
- [15] B. M. Chapman, B. Sundaram, K. Thyagaraja, "EZGrid system: A Resource broker for Grids," <http://www.cs.uh.edu/~ezgrid>
- [16] G. von Laszewski, I. Foster, J. Gawor, W. Smith, and S. Tuecke, "CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids," ACM 2000 Java Grande Conference, 2000.
- [17] J. Novotny, S. Tuecke, V. Welch, "An Online Credential Repository for the Grid: MyProxy," Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press, August 2001.
- [18] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, "A Security Architecture for Computational Grids," ACM Conference on Computers and Security, 1998, 83-91.
- [19] R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer, V. Welch, "A National-Scale Authentication Infrastructure," IEEE Computer, 2000.
- [20] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman, "Grid Information Services for Distributed Resource Sharing," 2001.
- [21] Resource Specification Language, RSL, http://www.globus.org/gram/rs1_spec1.html
- [22] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke, "A Resource Management Architecture for Metacomputing Systems," Proc.

- IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998.
- [23] B. Sundaram, B. M. Chapman, "Policy Engine: A Framework for Authorization, Accounting Policy Specification and Evaluation in Grids," 2nd International Conference on Grid Computing, Nov 2001.
 - [24] J. Boisseau, S. Mock, M. Thomas, "Development of Web Toolkits for Computational Science Portals: The NPACI HotPage", 9th IEEE Symposium on High Performance Distributed Computing, 2000
 - [25] Uniform Interface to Computing resource, UNICORE, <http://www.unicore.de>
 - [26] GridPort, <https://gridport.npaci.edu/>
 - [27] The Grid Portal Development Kit website is at <http://doesciencegrid.org/projects/GPDK/>
 - [28] HPCVL website is at <http://www.hpcvl.org>
 - [29] Entrust secure web portal, <http://www.entrust.com/solutions/webportal/>
 - [30] The source for Sun HPC ClusterTools can be downloaded from www.sun.com/solutions/hpc/communitysource
 - [31] Canada NRC-CBR, <http://cbr-rbc.nrc-cnrc.gc.ca/>
 - [32] Sun Center of Excellence program, <http://www.sun.com/products-n-solutions/edu/programs>
 - [33] SRS Sequence Retrieval System, <http://srs.ebi.ac.uk/>
 - [34] European Molecular Biology Network, EMBnet, <http://www.embnet.org/>
 - [35] Asian Pacific Bioinformatics Network, APBioNet, <http://www.apbionet.org/>
 - [36] ExPASy Molecular Biology Server, <http://us.expasy.org/>
 - [37] CANARIE, <http://www.canarie.ca/about/about.html>
 - [38] Cactus problem solving environment, <http://www.cactuscode.org/>
 - [39] C3.ca, Canadian High Performance Computing Collaboratory, <http://www.c3.ca/>
 - [40] Grid Canada, <http://www.c3.ca/>
 - [41] Network Information Service Plus, http://www.eng.auburn.edu/users/rayh/solaris/NIS+_FAQ.html
 - [42] EMBOSS, The European Molecular Biology Open Software Suite, <http://www.hgmp.mrc.ac.uk/Software/EMBOSS/>
 - [43] ClustalW sequence analysis, <http://www.ebi.ac.uk/clustalw/>
 - [44] Secure Shell, SSL, <http://www.openssh.com/>
 - [45] White Rose Grid, http://www.informatics.leeds.ac.uk/pages/05_facilities/06_grid.htm
 - [46] Sun ONE Studio, <http://www.sun.com/software/sundev/>
 - [47] Apache Tomcat server, <http://jakarta.apache.org/tomcat/>
 - [48] Further information on MyProxy can be found at <http://www.ncsa.uiuc.edu/Divisions/ACES/MyProxy/>
 - [49] PROGRESS project home page, <http://progress.psnc.pl/>
 - [50] PSNC Poznan Supercomputing and Networking Center, <http://www.man.poznan.pl/research/index.html>



A/Prof Simon See is currently the High Performance Computing Technology Director for Sun Microsystems Inc, Asia and also an Adjunct Associate Professor in Nanyang Technological University.

A/Prof See is also the director for the Sun Asia Pacific Science and Technology Center. His research interest is in the area of High Performance Computing, computational science, Applied Mathematics and simulation methodology. He has published over 50 papers in these areas and has won various awards.

Dr. See graduated from University of Salford (UK) with a Ph.D. in electrical engineering and numerical analysis in 1993. Prior to joining Sun, Dr See worked for SGI, DSO National Lab. of Singapore, IBM and International Simulation Ltd (UK). He is also providing consultancy to a number of national research and supercomputing centers. Dr See is also an adjunct research fellow in the National University of Singapore.

A Taxonomy and Survey of Grid Resource Planning and Reservation Systems for Grid Enabled Analysis Environment

Arshad Ali⁴, Ashiq Anjum⁴, Atif Mehmood⁴, Richard McClatchey³, Ian Willers²
Julian Bunn¹, Harvey Newman¹, Michael Thomas¹, Conrad Steenberg¹

¹California Institute of Technology, Pasadena, CA 91125, USA

Email: {conrad,newman}@hep.caltech.edu, Julian.Bunn@caltech.edu, thomas@hep.caltech.edu

²CERN, Geneva, Switzerland

Email: Ian.Willers@cern.ch

³University of the West of England, Bristol, UK

Email: Richard.mcclatchey@uwe.ac.uk

⁴National University of Sciences and Technology, Rawalpindi, Pakistan

Email: {arshad.ali, ashique.anjum, atif.mehmood}@niit.edu.pk

ABSTRACT

The concept of coupling geographically distributed resources for solving large scale problems is becoming increasingly popular forming what is popularly called grid computing. Management of resources in the Grid environment becomes complex as the resources are geographically distributed, heterogeneous in nature and owned by different individuals and organizations each having their own resource management policies and different access and cost models. There have been many projects that have designed and implemented the resource management systems with a variety of architectures and services. In this paper we have presented the general requirements that a Resource Management system should satisfy. The taxonomy has also been defined based on which survey of resource management systems in different existing Grid projects has been conducted to identify the key areas where these systems lack the desired functionality.

1. INTRODUCTION

Today, Grid users have to transform their high-level requirement into a workflow of jobs that can be submitted for execution on the Grid. Each job must specify which files contain the code to be run, selected by mapping the high level requirements to available application components and selecting a physical file from the many available replicas of the code in various locations. The job also specifies the location (or host) where it should be run, based on the code requirements (e.g., code is compiled for MPI, parallelized to run on tightly-coupled architecture, preferably with more than 5 nodes) and on user access policies to computing and storage resources. An executable workflow also includes jobs to move input data and application component files to the execution location.

Current Grid management systems allow the discovery of the available resources and data location but the users have to carry out all these steps manually. A resource planning and reservation system is thus required which can automate the whole process of work flow generation.

2. PLANNING AND RESERVATION

Planning and reservation is an important task to be performed by the Grid Resource Management System. Planning and Reservation is the process of analyzing the job and determining the resources required for successful execution of the job. Based on these results resources are reserved seamlessly to the user.

2.1 Requirements for planning and reservation

Resource management is a complex task involving security and fault tolerance along with scheduling. It is the manner in which resources are allocated, assigned, authenticated, authorized, assured, accounted, and audited. Resources include traditional resources like compute cycles, network bandwidth, space or a storage system and also services like data transfer, simulation etc.

Following are the requirements that a Grid RMS (Resource Management System) must satisfy in order to perform resource planning and reservation:

- A Grid RMS needs to schedule and control the resources on any element in the network computing system environment.
- Grid RMS should predict the impact that an application's request will have on the overall resource pool and quality of service guarantees already given to other applications.
- Grid RMS should preserve *site autonomy*. Traditional resource management systems work under the assumption that they have complete control on the resource and thus can implement the mechanisms and policies needed for effective use of that resource. But the Grid resources are distributed across separate administrative domains. This results in resource heterogeneity, differences in usage, scheduling policies and security mechanisms.
- Grid RMS must ensure Co-allocation of the resources. Co-allocation is the problem of allocating resources in different sites to an application simultaneously.
- Different administrative domains employ different local resource managements systems like NQE, LSF etc. A grid RMS should be able to interface and

interoperate with these local resource management systems.

- In a Grid system resources are added and removed dynamically. Different types of applications with different resource requirements are executed on the Grid. Resource owners set their own resource usage policies and costs. This necessitates a need for negotiation between resource users and resource providers so a grid RMS should enable such negotiation.
- The resource management framework should allow new policies to be incorporated into it without requiring substantial changes to the existing code.
- The Grid RMS is also responsible for ensuring the integrity of the underlying resource and thus enforces the security of resources. The resource management system must operate in conjunction with a security manager.

3. TAXONOMY

The taxonomy followed by us is based on the architecture of the planning and reservation system. Based on this taxonomy we have surveyed and classified various grid projects.

Different attributes in the taxonomy aim to differentiate RMS implementations according to the impact on overall Grid system scalability and reliability thus classification of RMS is based on grid type, resource namespace, resource information (discovery, dissemination), scheduling model and scheduling policy.

3.1 Grid Type

Grid systems are classified as Compute, Data and Service grids as shown in figure 2. The computational Grid category denotes the systems that have a higher aggregate computational capacity available for single applications than the capacity of any constituent machine in the system. The major resource managed by GRMS in compute grids is “Compute Cycles”.

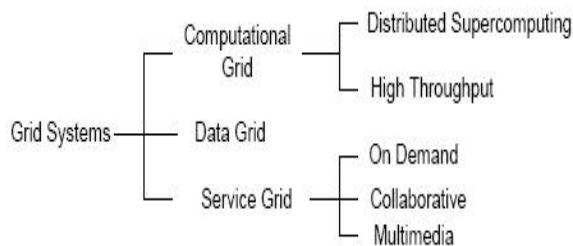


Figure 2 Data and Service grids

In Data Grids the resource management system manages data distributed over geographical locations. Data Grid is for systems that provide an infrastructure for synthesizing new information from data repositories such as digital libraries or Data Warehouses that are distributed in a wide area network. The Service Grid category is for the systems that provide services that are not provided by any single machine. This

category is further subdivided in On Demand, Collaborative, and Multimedia Grid Systems.

3.2 Resource Namespace

Resources in a grid are managed and named by the Grid Resource Management System; the naming of resources effects others functions of GRMS like resource discovery, resource dissemination and also affects the structure of the database storing resource information. Different approaches to naming are Relational, Hierarchical and graph based.

A relational namespace divides the resources into relations and uses the concepts from relational databases to indicate relationships between tuples in different relations.

A hierarchical namespace divides the resources in the grid into hierarchies organized around the physical or logical network structure of the grid i.e. it follows a system of systems approach, a name is constructed by traversing down a hierarchy.

In Graph Based Naming resources are linked together and a resource name is constructed by following the links from one object to another.

3.3 Resource Information

3.3.1 Resource Dissemination

Resource dissemination is the process of advertising information about resources. The protocols used for dissemination are “periodic” and “on demand”. In periodic resource dissemination the information database is updated periodically so update is not driven by resource status change indeed all changes are batched and updated in information database after specific interval. On Demand protocol updates the resource information database as the change occurs in the status of any of the resource.

3.3.2 Resource Discovery

Resource management system performs resource discovery to obtain information about available resources. There are two approaches to resource discovery namely “agent based” and “query based”. In Agent based approach agents traverse the grid system to gather information about resource availability. In Query Based approach resource information store is queried for resource availability.

3.4 Scheduling model

Scheduling model describes how machines involved in resource management make scheduling decisions. Scheduling models normally used are centralized and decentralized; in a centralized model all jobs are submitted to a single machine which is responsible for scheduling them on available resources. The problems with this approach are that the single scheduler will be single point of failure. It will also affect scalability of the grid. In decentralized model there is no central scheduler, scheduling is done by the resource requestors and owners independently. This approach is scalable and suits grid systems. But individual schedulers should cooperate with each other in making scheduling decisions.

3.5 Scheduling Policy

Scheduling policy governs how resources are scheduled on the matched resources. In a Grid environment there can be no single global scheduling policy, Different administrative domains may set different resource usage policies, so the RMS

should allow for the policies to be added or changed with minimal overhead.

4. SURVEY

Resource management in Condor, Globus, Legion, European Data Grid, and Nimrod G has been surveyed, keeping in view the above discussed taxonomy.

4.1 Planning and reservation system in Condor

The main function of Condor [4] is to allow utilization of machines that otherwise would be idle thus solving the wait-while-idle problem. Condor uses Classified Ads (which is a resource specification language) to specify resource requests. Through its unique remote system call capabilities, Condor preserves the job's originating machine environment on the execution machine, even if the originating and execution machines do not share a common file system and/or user ID scheme. Condor jobs with a single process are automatically checkpointed and migrated between workstations as needed to ensure eventual completion.

Condor has a centralized scheduling model. A machine (Central Manager) in the Condor system is dedicated to scheduling. Each Condor work station submits the jobs in its local queue to the central scheduler which is responsible for finding suitable resources for the job execution. The information about suitable available resources to run the job (execution machine information) is returned to the job submission machine. A shadow process is forked on the submission machine for each job, which is responsible for contacting and staging the job on the execution machine and monitoring its progress. Condor supports pre-emption of running jobs, if the execution machine decides to withdraw the resources Condor can preempt the job and schedule it on another machine thus providing for resource owner autonomy.

4.2 Planning and reservation system in Globus

Globus provides software infrastructure that enables applications to view distributed heterogeneous computing resources as a single virtual machine. The toolkit consists of a set of components that implement basic services, such as security, resource location, resource management, data management, resource reservation, and communications.

Planning and reservation system of Globus consists of *resource brokers*, *resource co-allocators* and *resource manager* or GRAM. The resource requests are specified in *extensible resource specification language (RSL)*.

Globus offers Grid information services via an LDAP-based network directory called Metacomputing Directory Services (MDS). The Resource Brokers discover resources by querying the information service (MDS) for resource availability. MDS consists of two components Grid Index Information service (GIIS) and Grid resource information service (GRIS). GRIS provides resource discovery services. GIIS provides a global view of the resources by pulling information from the GIIS's. Resource information on the GIIS's is updated by push dissemination.

Hierarchical name space organization is followed in Globus for naming resources and the scheduling model is decentralized i.e. scheduling is done by application level schedulers and resource

brokers. Co-allocator takes care of multi-requests, a multi request is a request involving resources at multiple sites which need to be used simultaneously, and passes each component of the request to appropriate resource manager and then provides a means for manipulating each resultant set of managers as a whole. The Co-allocation of resources is done by the DUROC component of Globus.

The resource manager interacts with local resource management systems to actually schedule and execute the jobs. The implementation of the resource manager in Globus is called GRAM. GRAM authenticates the resource requests and schedules them on the local resource manager. Each user is associated with a UHE (user hosting environment) on the execution machine. All the jobs from a user are directed to the user's UHE, which starts up a new Managed Job Factory service (MJFS) instance for every job.

The MJFS communicated with the clients by starting up two instances of File Stream Factory Service (FSFS) for standard input and output. MJFS and FSFS are persistent services.

4.3 Planning and reservation system in Legion

Legion [6] [9] is an operating system for the Grid that offers the infrastructure for Grid computing. Scheduler in Legion has a hierarchical structure. Users or active objects in the system invoke scheduling to run jobs, the higher level scheduler schedules the job on cluster or resource group while the local resource manager for that domain schedules the job on local resources. Scheduling in Legion is done by placing objects on the processors. The resource namespace is graph based.

Information about resources in the grid is stored in database object called a collection. For scalability there could be more than one collection object and collections can send and receive data from each other. Information is obtained from resources either by pull or push mechanism. Users or Schedulers query the collection to obtain resource information.

Legion supports resource reservation and object persistence. When the scheduler object contacts a host object (processor or local resource management system), the host returns a reservation token to the scheduler if the job can be executed on its resources.

Every object is associated with vault object. Vault object holds associated object's Object Persistent Representation (OPR). This ensures that even if the object fails, it can later be re-constructed from the OPR.

Communication between any two objects goes through the Legion Protocol stack which involves constructing program graphs, making method invocations, checking authorization, assembling or disassembling messages, encrypting, re-transmitting messages etc. This framework allows for implicit security and fault-tolerance

4.5. Planning and reservation system in European Data Grid

EU Data grid was designed to provide distributed scientific communities access to large sets of distributed computational and data resources. The main architecture of the datagrid is layered. The datagrid project develops datagrid services and depends on the Globus toolkit for core middleware services like

security. The datagrid services layer consists of workload management services which contain components for distributed scheduling and resource management, Data Management services contains middleware infrastructure for coherently managing information stores and monitoring services provided end-user and administrator access to status information on the grid. The workload management package consists of a user interface, resource broker, job submission service, book keeping and logging service. A job request from user is expressed in a Job Description Language based on the Classified Ads of Condor. The resource broker (RB) when given a job description tries to find the best match between the job requirements and available resources on the grid, considering also the current distribution of load on the grid. RB interacts with data replication and meta-data information services to obtain information about data location. The information service is LDAP based network directory. Resource discovery is done by queries and employ periodic push for dissemination. Global namespace hierarchical and scheduling is decentralized but instead of having a resource broker for each end-user, each virtual organization is provided resource broker. It does not support advanced reservation or co-allocation of resources. It does not address failures originated by jobs which it simply reports to end user. But the state of the resource broker queues and job submission service queues is persistent and can be recovered fully after a crash.

4.6. Planning and reservation system in Nimrod-G and GRACE

Nimrod-G [7] is a Grid grid-enabled resource management and scheduling system based on the concept of computational economy. It uses the middleware services provided by Globus Toolkit but can also be extended to other middleware services.

Nimrod-G uses the MDS services for resource discovery and GRAM APIs to dispatch jobs over grid resources. The users can specify the deadline by which the results of there experiments are needed. Nimrod-G broker tries to find the *cheapest* resources available that can do the job and meet the deadline. Nimrod uses both static cost model (stored in a file in the information database) and dynamic cost model (negotiates cost with the resource owner) for resource access cost trade-off with the deadline. GRACE provides middleware services needed by the resource brokers in dynamically trading resources access costs with the resource owners. It co-exists with other middle-ware systems like Globus. The main components of the GRACE infrastructure are Trade Manager (TM), trading protocols and Trade Server (TS). TM is the GRACE client in the Nimrod-G resource broker that uses the trading protocols to interact with trade servers and negotiate for access to resources at low cost. Trade Server is the resource owner agent that negotiates with resource users and sells access to resources. TS uses pricing algorithms as defined by the resource owner that may be driven by the demand and supply. It also interacts with the accounting system for recording resource usage.

It has an extensible application-oriented scheduling policy and scheduler uses theoretical and history based predictive techniques for state estimation. Scheduler organization is decentralized and the namespace is hierarchical.

5. CONCLUSION

In this paper various issues in resource planning and reservation have been discussed. A taxonomy based on architecture of grid resource management system has been described. Based on this taxonomy a survey of existing planning and reservation systems has been conducted and results are presented.

6. REFERENCES

1. Klaus Krauter, Rajkumar Buyya, and Muthucumaru Maheswaran, **A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing**, *International Journal of Software: Practice and Experience (SPE)*, ISSN: 0038-0644, Volume 32, Issue 2, Pages: 135-164, Wiley Press, USA, February 2002.
2. K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke. **A resource management architecture for Metacomputing systems**. In *Proceedings of the IPPS/SPDP Workshop on Job Scheduling Strategies for Parallel Processing*, pages 62–82, 1988.
3. Chaitanya Kandagatla : **Survey and Taxonomy of Grid Resource Management Systems**
4. Condor Team. Condor Manual. Available from <http://www.cs.wisc.edu/condor/manual>, 2001.
5. Condor Team. The directed acyclic graph manager. <http://www.cs.wisc.edu/condor/dagman>, 2002.
6. H. Dail, G. Obertelli, F. Berman, R. Wolski, and Andrew Grimshaw, **Application-Aware Scheduling of a Magnetohydrodynamics Application in the Legion Metasystem**, *Proceedings of the 9th Heterogeneous Computing Workshop*, May 2000.
7. R. Buyya, D. Abramson, J. Giddy, **Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid**, *International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, Beijing, China. IEEE Computer Society Press, USA, 2000.
8. W. Hoschek, J. Jaen-Martinez, A. Samar, H. Stockinger, and K. Stockinger, **Data Management in an International Data Grid Project**, *Proceedings of the first IEEE/ACM International Workshop on Grid Computing*, (Springer Verlag Press, Germany), India, 2000.
9. S. Chapin, J. Karpovich, A. Grimshaw, **The Legion Resource Management System**, *Proceedings of the 5th Workshop on Job Scheduling Strategies for Parallel Processing*, April 1999.

A Comparative Survey of Fault-tolerant and Load balanced MPI Implementations, Software Packages and Algorithms *

Raihan Ur Rasool and Guo Qingping
School of Computer Science and Technology
The Wuhan University of Technology
Wuhan 430063, Hubei, China
Email: {qpguo, Raihan}@mail.whut.edu.cn

ABSTRACT

Advancement in high-speed networks and rapid improvement in microprocessor design enabled cost-effective high-performance parallel computing on clustered low cost workstations and PCs. Systems based on message passing draw attractions in the field of high performance computing, where loop or data parallelism is a main source of parallel processing. Therefore, MPI is one of the most adopted programming models for Large Clusters and Grid deployments. However, these systems often suffer from network or node failures, because the nodes of such systems are likely to be heterogeneous with respect to computing power and workload. This raises the issue of selecting a fault tolerance approach for MPI to manage various dynamic failures appropriately. Moreover, loads should be balanced according to the performance of nodes to minimize the elapsed time of program. Various models have emerged to simplify the task of programming in network environment, but MPI approach is considered one of the most mature methods currently used in parallel programming. This paper presents an overview of Message-passing models in the context of Grid Computing and a comparative survey of fault-tolerant and load balanced system, algorithms and software packages for MPI.

Keywords: Fault-tolerant MPI, Grid Computing, Dynamic Load-balanced systems, Parallel Computing, Cluster of workstations

1. INTRODUCTION

Driven by increasingly complex problems and propelled by increasingly powerful technology, today's science is as much based on computation, data analysis, and collaboration as on the efforts of individual experimentalists and theorists. But even as computer power, data storage, and communication continue to improve exponentially, computational resources are failing to keep up with what scientists demand of them. A personal computer in 2001 is as fast as a supercomputer of 1990. A useful metric for the rate of technological change is the average period during which speed or capacity doubles or, more or less equivalently, halves in price. For storage, networks, and computing power, these periods are around 12, 9, and 18 months, respectively. The different time constants associated with these three exponentials have significant implications.

Employing clusters of workstations as a cost effective alternative to parallel computers has been the goal of much

research in the past few years [1], [2], especially in light of the remarkable advances in both computing power of PCs and networks speed. As the performance of computer networks improves and the software supports of interprocess communication such as PVM [3] and MPI [4] prevail, multicomputers based on message passing emerge as a viable platform for high performance computing. Multicomputer has a wide spectrum of systems from MPP to a cluster of workstations. The idea to build such clusters is very appealing, but it is too complicated to realize because of several issues.

For systems gathering thousands of nodes, node failures or disconnections are not rare, but frequent events. For Large Scale Machine like the ASCI-Q machine, the MTBF (Mean Time Between Failure) for the full system is estimated to few hours. The Google Cluster using about 8000 nodes experiences a node failure rate of 2-3% per year [5]. This can be translated to a node failure every 36 hours.

The MPI standard has proven effective and sufficient for high-performance applications, in situations without either QoS or fault-handling requirement. However a main drawback of message passing is its high communication overhead, which includes software overhead, hardware latency and delay caused by network and memory contention [6].

When exploiting parallelism on multicomputers, we confront several challenging problems regardless of the form of workload. The first problem is load-balancing. In a shared running environment, the working condition of nodes changes dynamically and unpredictably due to the interference of the operating system and other processes. Therefore, any load balancing algorithms for multicomputers should be adaptive to heterogeneity of processor speed, network latency and workload. Balancing the runtime computational load, is usually very difficult due to several reasons. These include a reliable measurement of the computational load, the amount of runtime data movement, and the minimization of interprocessor communication. Various methods on dynamic load balancing have been reported to date by numerous researchers; however, most of them lack a global view of loads across processors.

In addition, a multicomputer system, especially NOW/COW, may experience various dynamic failures. There are several ways to implement fault tolerance in MPI: a) the programmer of the application may save periodically intermediate results on reliable media during the execution in case of global restart, b) the functions of the MPI implementation may return fault notification information and accept reconfiguration of the communication context and c) the MPI implementation provides a fully automatic fault detection and transparent recovery. The automatic approach suffers either of limited fault tolerance capabilities or high resource cost. A robust algorithm should deal with failures under a dynamic

* This work is supported by the Natural Science Foundation of China (NSFC No. 60173046)

environment.

The second section of the paper presents a survey of Message-passing models for parallel computing. Section 3 presents the overview of some fault tolerant MPI implementations. Section 4 offers a deep look into the widely used load balanced software packages and algorithms.

2. MESSAGE-PASSING MODELS

In message-passing models, processes run in disjoint address spaces, and information is exchanged using message passing of one form or another. While the explicit parallelization with message passing can be cumbersome, it gives the user full control and is thus applicable to problems where more convenient semiautomatic programming models may fail. It also forces the programmer to consider exactly where a potential expensive communication must take place. The principal of message passing rests on tasks cooperation through explicit message exchanges carried out as point-to-point communication between two processes or between several processes and a unique communication task.

2.1 MPI and variants

The Message Passing Interface (MPI) [7] is a widely adopted standard that defines a two-sided message passing library, that is, with matched sends and receives, that is well-suited for Grids. The most important feature of MPI is its support for modular programming. A communicator allows the MPI programmer to define modules that encapsulate internal communications structures [8]. Many implementations and variants of MPI have been produced namely MPICH-G2, FM-MPI, WMPI, MPI/PRO, PACX-MPI, MPI Connect, MagPIe library and PaTENT. The most prominent for Grid computing is MPICH-G2.

MPICH-G2: MPICH-G2 [9] is a Grid-enabled implementation of the MPI that uses the Globus services (e.g. job start-up, security) and allows programmers to couple multiple machines, potentially of different architectures, to run MPI applications. MPICH-G2 automatically converts data in messages sent between machines of different architectures and supports multiprotocol communication by automatically selecting TCP for intermachine messaging and vendor-supplied MPI for intramachine messaging. MPICH-G2 requires, however, that Globus services be available on all participating computers to contact each remote machine, authenticate the user on each, and initiate execution (e.g. fork, place into queues, etc.).

FM-MPI: FM-MPI is a version of MPICH built on top of Fast Message. The FM (Fast Message) interface is based on Berkeley Active Message. The FM interface was originally developed on Cray T3D and a cluster of SPARC stations connected by Myrinet. Recently, a variant of FM-MPI that runs on top of WinSock 2 was released as part of the High-Performance Virtual Machines (HPVM) project being undertaken by CSAG [10], [11]

MagPIe: The MagPIe library [12] implements MPI's collective operations such as broadcast, barrier, and reduce operations with optimizations for wide-area systems as Grids. Existing parallel MPI applications can be run on Grid platforms using MagPIe by relinking with the MagPIe library. MagPIe has a simple API through which the underlying Grid

computing platform provides the information about the number of clusters in use, and which process is located in which cluster.

MPI/Pro: MPI/Pro is a commercial environment released in April 1998 by MPI Software Technology, Inc. MPI/Pro is based on WinMPIch. The current version of MPI/Pro is fairly radically redesigned to remove the bottlenecks and other problems that were present. The MPI/Pro developers are currently working on new source based for MPI that does not include any MPICH code and supports the Virtual Interface (VI) Architecture [13]. MPI/Pro provides multi-device architecture that allows MPI applications to efficiently exploit SMP parallelism; multithreaded design; user level thread safety; asynchronous method of synchronization and notification and optimized derived data types [14]

PACX-MPI: PACX-MPI [15] has improvements for collective operations and support for intermachine communication using TCP and SSL. Stampi has support for MPI-IO and MPI-2 dynamic process management.

MPI Connect: MPI Connect [16] enables different MPI applications, under potentially different vendor MPI implementations, to communicate.

PaTENT MPI: PaTENT MPI 4.0 is a high performance implementation of MPI on Windows NT showing outstanding performance and robustness in commercial environments. It is the commercial version of WMPI funded by EU project WINPAR. It is communication library and run-time system designed to facilitate the parallelization of numerical codes to run on multiple CPUs and workstations across a network. It also offers a full standard MPI implementation for Microsoft Win32 platforms based on Windows NT workstation and server clusters. This is the first software component of PaTENT, the soon-to-be releases suit of NT parallel Tools Environment to be used for the development of parallel application software [17]. PaTENT MPI 4.0 can co-exists and co-operate over a TCP/IP network with UNIX based implementations

3. FAULT TOLERANT IMPLEMENTATIONS AND MODELS

The dynamic nature of clusters of workstations and Grid means that some level of fault tolerance is necessary. This is especially true for highly distributed codes such as Monte Carlo or parameter sweep applications that could initiate thousands of similar, independent jobs on thousands of hosts. Clearly, as the number of resources involved increases, so does the probability that some resource will fail during the computation. Here we present a survey of some distributed fault-tolerant implementations and architectures.

Manetho: Manetho is a distributed fault-tolerance implementation that runs on a cluster of workstations [18]. It uses causal message logging to provide for system recovery. Because a Manetho process logs both the data of the messages that it sends and the non-deterministic events that these messages depend on, the size of those logs may grow very large if used with a program that generates a high volume of large messages, as is the case for many scientific programs. While Manetho can bound the size of these logs by occasionally checkpointing process state to disk, programs

that perform a large amount of communication would require very frequent checkpointing to avoid running out of log space. Furthermore, since the system requires a process to take a checkpoint whenever these logs get too large, it is not clear how to use this approach in the context of application-level checkpointing. Moreover Manetho was not designed to work with any standard message passing API, and thus does not need to deal with the complex constructs – such as non-blocking and collective communication – found in MPI. This system uses a novel combination of rollback-recovery and process replications to provide fault tolerance and high availability; it uses process replication to provide high availability to servers in the system [19].

Condor: Condor [20] is a software package for executing batch jobs on a variety of UNIX platforms, in particular, those that would otherwise be idle. It is actually a distributed system that runs on a cluster of workstations. The major features of Condor are automatic resource location and job allocation, checkpointing, and the migration of processes. These features are implemented without modification to the underlying UNIX kernel. However, it is necessary for a user to link their source code with Condor libraries. Condor monitors the activity on all the participating computing resources; those machines that are determined to be available are placed in a resource pool. Machines are then allocated from the pool for the execution of jobs. The pool is a dynamic entity – workstations enter when they become idle and leave when they get busy. Condor provides an environment for executing serial and parallel applications on clusters. Moreover, it supports checkpoint/restart in order to provide fault tolerance and process migration [21].

Legion: Legion is an object-based meta-system [23]. It has been built on a collection of connected hosts to provide a virtual computer that can access all types of data and physical resources. Legion is designed to be a worldwide virtual computer. Legion [22] provides objects with a globally unique (and opaque) identifier. Using such an identifier, an object, and its members, can be referenced from anywhere. Being able to generate and dereference globally unique identifiers requires a significant distributed infrastructure. We note that all Legion development is now being done as part of the AVAKI Corporation.

MPI/FT: MPI/FT expands MPI in novel way to include scope for fault/error detection and recovery. The MPI [3] standard requires that successful completion of an MPI application imply that all processes complete successfully, and the default behavior in case of process failure is the immediate termination of application. MPI-1.2 allows users to attach an error handling function to each communicator, which would be invoked in case of an abnormal return. However, performance constraints prevent MPI from detecting certain errors, and “catastrophic errors may prevent MPI calls from returning to the caller, thereby preventing invocation of the user error handler. MPI/Pro, MPICH and LAM are some of the existing implementations of MPI, and none currently address fault issues. MPI/FT trades off sufficient MPI fault coverage against acceptable parallel performance, based on mission requirements and constraints. MPI codes are evolved to use MPI/FT features. Non-portable code for event handlers and recovery management is isolated. Process, node and network failure are some of the faults that present themselves. MPI/FT acts as the middleware/tool that incorporates many inaccuracies of computed results [39].

Starfish: Starfish supports several approaches to checkpoint/restart and employs group communication for providing fault-tolerance and high availability. In addition, starfish allows dynamic changes in the number of running processes. It is an environment for executing dynamic (and static) MPI 2 programs on a cluster of workstations. Starfish is efficient, fault tolerant, highly available and dynamic as a system internally, and in supporting fault-tolerance and dynamicity for its application programs as well. Starfish achieves these goals by combining group communication technology with checkpoint/restart, and uses a novel architecture that is both flexible and keeps group communication outside the critical data path, for maximum performance [24].

Cocheck: Cocheck [23] is among the earliest efforts towards incorporating a limited fault tolerance capability in MPI. Cocheck provides only the functionality for the coordination of distributed checkpoints, relying on the Condor system to take system-level checkpoints of each process. It is a checkpointing library layered over MPI, and is an extension of the single process checkpoint of Condor [11] with a protocol for synchronous checkpointing. This causes all the messages in transit to be flushed to arrive at a consistent global state after which the application is checkpointed. Cocheck uses a coarse-grain approach and is primarily developed and optimized for process migration. Limitations include scalability issues because of the control process and the high overhead associated with the flush protocol. Cocheck cannot consequently provide effective transient faults coverage.

MPI/RT: The MPI/RT [25] standard is designed specifically to address issues related to Quality of Service (QoS), resource management and scheduling of communication tasks, which are not addresses in the other MPI's. Though the error handling capability of MPI/RT is much more sophisticated than MPI-1.x and 2.x, it is still inadequate for fault-tolerance purpose, notably useful to fault-tolerance is the Dynamic Process Management (DPM) capability of MPI-2 [1]. DPM is however, insufficient to handle failures such as process crashes.

MPICH-V: MPICH-V is a distributed, asynchronous automatic fault tolerant MPI implementation designed for large-scale clusters, Global and Peer-to-Peer Computing platforms. It is an automatic Volatility tolerant MPI environment based on uncoordinated checkpoint/rollback and distributed message logging. MPICH-V architecture relies on Channel Memories, Checkpoint servers and theoretically proven protocols to execute existing or new, SPMD and Master-Worker MPI applications on volatile nodes. MPICH-V environment encompasses a communication library based on MPICH [26] and a runtime environment. The MPICH-V library can be linked with any existing MPI program as usual MPI libraries. The library implements all communication subroutines provided by MPICH. Its design is a layered architecture: the peculiarities of the underlying communication facilities are encapsulated in a software layer called a Device', from which all the MPI functions are automatically built by the MPICH compilation system. The MPICH-V library is build on top of a dedicated device ensuring a full-fledged MPICH v.1.2.3., implementing the Chameleon-level communication functions. The underlying communication layer relies on TCP for ensuring message integrity. MPICH-V relies on the concept of Channel Memory

(CM) to ensure fault tolerance and to allow the firewall bypass [27].

Libckpt: Libckpt is a transparent check pointing library on uniprocessors running UNIX [28]. It provides a mechanism for enabling fault-tolerance for long-running programming. Libckpt implements most optimizations that have been proposed to improve the performance of check pointing, including, e.g., incremental checkpointing, forked check pointing, and copy-on-write checkpointing [29].

4. LOAD BALANCING SYSTEMS AND ALGORITHMS

4.1 Load Balancing Systems

Below we have given details about three load balancing software packages after extensive examination. This analysis and examination allows us to determine which characteristics are of the most benefit when load balancing asynchronous and irregular applications.

4.1.1 ParMETIS: Repartitioning tools are the most frequently used dynamic load balancing methods found in the scientific computing literature. These methods make use of a priori knowledge of the computation in order to partition the workload into a user-specified number of chunks. Some methods use graph partitioning algorithms to divide an initial graph into equally weighted subgraphs. Other methods are more application-specific, and may choose to optimize certain criteria, such as subdomain surface-to-volume ratio, cut edge weights, or data redistribution costs. Repartitioning tools can be found incorporated into such projects as Jostle, DRAMA, Zoltan, and Metis. Two common methods exist for creating a new partitioning for an already distributed mesh that has become load imbalanced due to mesh refinement and coarsening: scratch-remap schemes create an entirely new partition and tend to more evenly distribute load, while diffusive schemes attempt to tweak the existing partition to achieve better load balance, often minimizing data migration costs. Metis' ParMETIS V3 AdaptiveRepart () routine makes use of a Unified Repartitioning Algorithm [30], which combines the characteristics of both scratch-remap and diffusive schemes.

A parameter known as the Relative Cost Factor is application-defined and describes the relative costs required for performing interprocessor communication during parallel processing and performing data redistribution associated with load balancing. This gives rise to the minimization function

$$|E_{cut}| + \alpha |V_{move}| \quad (1)$$

Where $|E_{cut}|$ is the edge-cut of the partitioning, and $|V_{move}|$ is the total cost of data redistribution. Repartitioning progresses in three stages. First, the graph is coarsened using a local variant of heavy-edge matching that is shown to be effective at helping to minimize both the number of edge-cuts and data redistribution costs. In addition, this algorithm is scalable to a large number of processors. The second step is to create an initial partition. Because the most beneficial method depends on the particular problem instance, as well as the value chosen for the Relative Cost Factor (α), the initial partition is created twice (once using a scratch-remap method, and once using a diffusive method). Finally, a multilevel refinement algorithm is used while minimizing Equation 1 [31].

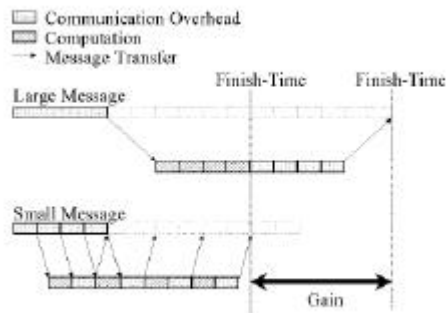
4.1.2 Charm ++: In many cases, applications (e.g. simulations) are organized as a series of discrete time steps. In such cases, it is often beneficial to perform load balancing at strategic locations, rather than at arbitrary points during the computation. Charm++ [32] provides a runtime framework in which load balancing policies may be “plugged into” an application in a modular fashion. With each module provided in the Charm++ distribution, the load balancing methods are implemented using a global barrier, making them well suited for loosely synchronous computations¹. Charm++ presents a programming model in which the application data domain is divided into a number of chunks, with the number of chunks being much greater than the available number of physical processors. Each chunk is represented as a chare object, whose interface is defined by entry point methods. Messages invoke computation by specifying the entry point to execute upon reception. Load balancing is achieved by mapping and re-mapping chares to available processors. An assumption, known as the principle of persistent computation and communication structure [33], is made which states that changes to the computation and communication structure of an application happen slowly or infrequently. Two components make up the Charm++ load balancing framework: the specific load balancing policy or strategy and a distributed load balancing database constructed through runtime monitoring of the application. The load-balancing module makes use of the information contained within the database (possibly gathering it at a central location, if necessary) to determine what chares should migrate in order to balance the runtime load. Because creating an optimal load distribution is an NP-hard problem that involves optimizing for both interprocessor communication and load distribution. The simplest are Greedy Strategies among several heuristic approaches, which sort both chare workloads and processor load levels in order to assign the heaviest free chare to the processor with the lightest current load. Such a strategy may result in a large amount of data migration. Refinement Strategies aim to minimize the number of chare migrations while improving load balance. For each overloaded processor only, heavy objects are migrated to underloaded processors until the load falls below a threshold, which is defined as a percentage of the average processor workload [31].

4.1.3 PREMA: The Parallel Runtime Environment for Multicomputer Applications (PREMA) is a runtime library based on a design philosophy which includes: a single-sided communication [34] similar to what is provided by Active Messages, a global namespace which assigns a unique identifier to application-defined mobile objects, transparent object migration and automatic message forwarding for mobile objects, a framework which allows for the easy and efficient implementation of customized dynamic load balancing algorithms and a suite of commonly used dynamic load balancing strategies, such as Diffusion [35] and Multi-list Scheduling [36]. The PREMA library allows load balancing to be initiated either explicitly or implicitly. The application may explicitly hand control to the load balancer by posting a polling operation, which will check for incoming application messages, schedule the next work unit for execution, evaluate the current local work level, and process any system-generated load balancing messages. Alternatively, the runtime system may preempt the application at periodic intervals and perform load-balancing functions. Note that, even in the case of preemptive load balancing, it is still necessary for the application to poll for its own messages. [31]

4.2 Load Balancing Algorithms and Strategies

Load distribution in distributed environments has been a challenging issue in the parallel computing society. Regardless of the form of workload being task or data, load balancing algorithms can be categorized according to their characteristics. First, algorithms can be classified as centralized and decentralized. Centralized algorithms have one special purpose node called coordinator or master, which manages global load information and distributes loads according to the information. However, in case of decentralized algorithms, there is no coordinator, but each node manages its local (global) load information and balances the workload according to the information interacting with other nodes. In addition, algorithms can be characterized as relocatable and irrelocatable. Relocatable methods distribute loads at an early stage of distribution, and monitor the workloads of nodes. When the degree of load imbalance exceeds a certain threshold, an algorithm directs loads to migrate from a heavily loaded node to a lightly loaded one. On the other hand, irrelocatable schemes try to schedule or partition data without a significant load imbalance. As irrelocatable schemes have no mechanism of load migration, the scheduling should anticipate the performance of nodes accurately [37].

4.2.1 RAS Load Balancing Algorithm: RAS [37] solves the load-balancing problem and dynamic failures by a work stealing mechanism, and the processor selection problem by data distribution based on a reservation scheme. To reduce or hide the communication overhead, RAS adopts an overlapped communication. In message passing architectures, sending a large message is more desirable than multiple small messages, because it amortizes the high start-up cost.



However, as shown in Figure, a large message delays the slaves' start-time. In order to achieve high performance, there has to be a trade-off between the start-up cost and the start-time. The communication overhead is proportional to the size of message, if the size is large enough. We can find the point where linearity is broken; the message larger than 1KB gives us relatively constant overhead per unit message. As a result, a good compromise is to use multiple small messages of size larger than 1KB. RAS allows several slaves to redundantly compute a data. This mechanism makes RAS survive under a dynamic node failure [37].

4.2.2 Load-Balancing Scatter Operations: The *scatter* operation consists of distributing n pieces of data initially held by one processor (the *source*) among the n processors making up the parallel system, including the source. The n pieces of data are indexed with natural numbers from 0 to $n-1$ as the processors are, and the scatter requires the piece i to be sent to

processor p_i . The typical usage of the scatter operation is to spawn an SPMD computation section on the processors after they received their piece of data. Thereby, if the computation load on processors depends on the data received, scatter operation can be used as a means to load-balance computations, provided the items in the data set to scatter are independent. MPI provides the primitive `MPI_Scatterv` that allows distributing *unequal* shares of data. It is claimed here that replacing `MPI_Scatter` by `MPI_Scatterv` calls parameterized with clever distributions may lead to great performance improvements at low cost. In term of source code rewriting, the transformation of such operations does not require a deep source code re-organization, and it can easily be automated in a software tool. In this strategy the problem which is tried to solve is to statically load-balance the execution by computing a data distribution depending on the processors speeds and network links bandwidths [38].

5. CONCLUSION

To write a portable and efficient parallel program on Clusters of workstations, user must take into account three problems: load balancing, processor selection and dynamic node failures. Clusters of workstations offer a potential for cost effective high-performance computing. However, building usable clusters is inherently difficult task. Successful implementations of such clusters must retain high-performance, while addressing issues like manageability, fault-tolerance, high-availability, and coping with dynamic changes in the environment.

In this paper we partially addressed the problem of Fault-tolerance and Load-balancing. We studied Message-passing models in the context of Grid Computing and presented a survey of fault-tolerant and load balanced system, algorithms and software packages for MPI.

To conclude, large scale clusters and Grid systems raise the issue of tolerance to frequent and numerous faults. Since these systems are mostly programmed using MPI, the use of a fault tolerant and load balanced MPI implementation will become unavoidable. Hence, it is required to build up a user friendly library easy to develop parallel programs without the knowledge of the issues in detail. In addition to be tolerant to dynamic failures and load balancing, we need more sophisticated error handling routines.

6. REFERENCES

- [1] T. Anderson, D. Culler, and D. Patterson. A Case for NOW (Network of Workstations). *IEEE Micro*, February 1995.
- [2] A. Chien, M. Lauria, R. Pennington, M. Showerman, G. Iannello, M. Buchanan, K. Hane, L. Giannini, G. Koenig, S. Krishnamurthy, Q. Liu, S. Pakin, and G. Sampemane, The Design and Evaluation of an HPVM-based Windows-NT Supercomputer. Unpublished manuscript, 1999.
- [3] Beguelin, J. Dongarra, A. Geist, R. Manchek, and V. Sunderam. "A Users' Guide to PVM (Parallel Virtual Machine)". Technical Report ORNL/TM-11826, Oak Ridge National Laboratory, 1991.
- [4] *MPI: A Message-Passing Interface Standard*. Message Passing Interface Forum, May, 1994.
- [5] A. Brown and D. A. Patterson. Embracing failure: A case for recovery-oriented computing (roc). In *High Performance*

- Transaction Processing Symposium, Asilomar, CA, October 2001.
- [6] Zhiwei Xu and Kai Hwang. "Modeling Communication Overhead: MPI and MPL Performance on the IBM SP2". *IEEE Parallel and Distributed Technology*, Spring, pp.9-23, 1996.
 - [7] Message Passing Interface Forum, MPI: A Message Passing Interface Standard, June, 1995, www.mpi-forum.org/.
 - [8] Ian Foster. Designing and Building Parallel Programs :Concepts and Tools for parallel Software Engineering, Addison-Wesley, New York, 1995.
 - [9] Foster, I. and Karonis, N. T. (1998) A grid-enabled MPI: message passing in heterogeneous distributed computing systems. *Supercomputing*. IEEE, November, 1998, www.supercomp.org/sc98
 - [10] HPVM- <http://www.cs.gsu.edu/projects/clusters.html>
 - [11] S. Parkin, M. Lauria, A.Chein, et. Al, " High Performance Virtual Machines (HPVM): Clusters with Supercomputing APIs and Performance", Eighth SIAM Conference on Parallel Processing for Scientific Computing (PP97); March, 1997
 - [12] Kielmann, T., Hofman, R. F. H., Bal, H. E., Plaat, A. and Bhoedjang, R. A. F. (1999) magPIe: MPI's collective communication operations for clustered wide area systems. *Proc. Seventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'99)*, Atlanta, GA, May 4-6, 1999 pp. 131-140.
 - [13] VIA- <http://www.via.com/>.
 - [14] Dimitrov, R. and A. Skjellum, " Efficient MPI for Virtual Interface (VI) Architecture ", *Proceedings of 1999 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, USA, June 1999, Vol.6,pp:3094-3100*
 - [15] Gabriel, E., Resch, M., Beisel, T. and Keller, R. (1998) Distributed computing in a heterogeneous Computing Environment, in Alexandrov, V. and Dongarra, J. (eds) *Recent Advances in Parallel Virtual Machine and Message Passing Interface. 5th European PVM/MPI Users' Group Meeting*, Springer, 180-188, Liverpool, UK.
 - [16] Fagg, G. E., London, K. S. and Dongarra, J. J. (1998) MPI Connect: managing heterogeneous MPI applications interoperation and process control, in Alexandrov, V. and Dongarra, J. (eds) *Recent Advances in Parallel Virtual Machine and Message Passing Interface. Lecture Notes in Computer Science, Vol. 1497*. Springer, pp. 93-96, *5th European PVM/MPI Users' Group Meeting*, Liverpool, UK.
 - [17] PaTENT MPI 4.0- <http://www.genias.de/product/patent/index.html>
 - [18] E. N. Elnozahy. Manetho: Fault Tolerance in Distributed Systems Using Rollback-Recovery and Process Replication. Phd thesis, Houston University, October 1993.
 - [19] K.Birman, The Process Group Approach to *Reliable Distributed Computing*. *Communications of the ACM*, 36(12):37-53, December 1993.
 - [20] Condor, <http://www.cs.wisc.edu/condor/>.
 - [21] M. Litzkow, T. Tannenbaum, J. Basney, and M. Livny. Matchmaking: Distributed resource Management for High Throughput Computing. Technical Report 1346, University of Wisconsin-Madison Computer Sciences, April 1997.
 - [22] Lewis, M. and Grimshaw, A. (1995) The Core Legion Object Model, Technical Report TR CS-95-35, University of Virginia, 1995.
 - [23] A. Basu, V. Buch, w. Vogels, and T. von Eiken. UNet: A User-Level Network Interface for Parallel and Distributed Computing. In *Proc. of the 15th ACM Symposium on Operating Systems Principles*, pages 40-53, December 1996.
 - [24] Adnan M: agbaria and Roy Friedman. Starfish: Fault-Tolerant Dynamic MPI Programs on Clusters of Workstations. Department of Computer Science, The Technion Haifa 32000 Israel
 - [25] Y.Amir, L. E. Moser, P. M. Melliar-Smith, D. Agarwal, and P. Ciarfella. Fast Message.rdering and Membership Using a Logical Token-Passing Ring. In *Proc. of the 13th International Conference on Distributed Computing Systems*, pages 551-560, May 1993
 - [26] William Gropp, Ewing Lusk, Nathan Doss, and Anthony Skjellum. High-performance portable implementation of the MPI Message Passing Interface Standard. *Parallel Computing*, 22(6):789-828, September 1996..eorge Bosilca, Aurelien Bouteiller and Franck Cappello, MPICH-V: Toward a Scalable Fault Tolerant MPI for Volatile Nodes. LRI, Universit'e de Paris Sud, Orsay, France.
 - [27] J. S. Plank, M. Bech, G. Kingsley, and K. Li. Libckpt: Transparent Checkpointing UnderUNIX. In Usenix Winter 1995 Technical Conference, pages 220-232, New Orleans, January 1995
 - [28] J .S. Plank, An Overview of Checkpointing in Uniprocessor and Distributed systems,Focusing on Implementation and Performance. Technical Report UTCS-97-372, Department of Computer Science, Tennessee Univeristy, July 1997
 - [29] K. Schloegel, G. Karypis, and V. Kumar. A unified algorithm for load-balancing adaptive scientific simulations. In *Proc. of the Intl. Conf. on Supercomputing*, 2000.
 - [30] Kevin J. Barker and Nikos P. Chrisochoides. An Evaluation of a Framework for the Dynamic Load Balancing of Highly Adaptive and Irregular Parallel Applications, Dept. of Computer Science College of William and Mary Williamsburg, VA 23185
 - [31] L. Kale and S. Krishnan. CHARM++: A portable concurrent object oriented system based on C++. In *Proceedings of OOPSLA '93*, pages 91{108, 1993.
 - [32] M. Bhandarkar, L. Kale, E. Sturler, and J. Hoeflinger. Object-based adaptive load balancing for mpi programs. *Technical Report 00-03, Univ. of Illinois at Urbana-Champaign*, 2000.
 - [33] K. Barker, N. Chrisochoides, J. Dobbelaere, and D. Nave. Data movement and control.ubstrate for parallel adaptive applications. *Concurrency Practice and Experience*, 14:77{101, 2002.
 - [34] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Par. And Dist. Comp.*, 7(2):279{301, 1989.
 - [35] I. Wu. *Multilist Scheduling: A New Parallel Programming Model*. PhD thesis, School of Comp. Sci., Carnegie Mellon University, Pittsburg, PA 15213, July 1993.
 - [36] Yangsuk Kee and Soonhoi Ha: A Robust Dynamic Load-Balancing Scheme for Data.arallel Application on Multicomputer Systems.
 - [37] Stephane Genuad and Arnaud Giersch. Load-Balancing Scatter Operations for GridComputing. *IEEE Processdings of the International Parallel and distributed Processing Symposium*. Strasboug
 - [38] Rajanikanth Batchu, Jothi P. Neelamegam. MPI/FT: Architecture and Taxonomies for Fault-Tolerant, Message-Passing Middleware for Performance-Portable Parallel Computing. Department of Computer Science HPC Laboratory, Starkville, USA

Reusing Legacy Applications for Grid Computing *

Yu Huashan, Xu Zhuoqun, Ding Wenkui

School of Electronic Engineering and Computer Science, Peking University

Beijing, 100871, P.R. China

Email: yuhs@ailab.pku.edu.cn Tel: 010-62754248

ABSTRACT

In scientific computation domains, there are a large number of legacy applications that run on MPPs, clusters and workstations for daily work. One single application alone is generally restricted in computability and cannot meet the requirements of modern scientific problems. This paper presents a component model AOD for coordinating them to solve complex problems on the computational grid. Based on the software component technologies, legacy applications and their target platforms are encapsulated to be grid-programming components on the computational grid. Every grid-programming component provides a set of domain-termed operators that can be referred in grid applications to perform specific computation. A grid-programming component also encapsulates the domain-specific expertise for implementing its operators with the encapsulated resources. The AOD represents every grid application as an acyclic and directed graph that describes a workflow of references to operators provided by grid-programming components. These referred grid-programming components are invoked and coordinated by the AOD at runtime. We have developed a prototype of AOD, and an experimental result is presented to evaluate the implementation.

Keywords: Computational Grid, Software Component, Legacy application, Concurrency.

1. INTRODUCTION

A primary goal of Grid computing [1,2] is to coordinate a large collection of heterogeneous resources for sharing universally, so as to create a dependable and efficient computing platform for large-scale complex scientific problems. To realize this goal, applications are to schedule and combine distributed resources dynamically, according to factors such as availability of grid resources and requirements of submitted jobs. Among the myriad resources are existent applications that run independently on local platforms such as clusters, MPPs and workstations. They were separately developed with different programming languages and models. Generally, these legacy applications were artfully designed and extensively optimized to meet some specific domain requirements. Some of them are overlapping or complimentary in respect of function. Reusing them for grid computing has many potential advantages. For example, in order to improve efficiency and resolvable problem sizes, multiple legacy applications can be composed to be an innovative application running on the computational grid. At runtime, components of the innovative application are scheduled to run concurrently on different local platforms, and each component completes a piece of computation.

Although reusing legacy applications for grid computing holds the promise to simplify the complexity of developing grid applications and support innovative computation, it poses several challenges for both programmers and the computational grid, including balancing workload universally, interoperating between heterogeneous resources, and synchronizing distributed resources, etc. This work aims to develop an architecture that allows legacy applications augmented with syntactic and semantic information to be incorporated into the computational grid, so as to be shared universally and transparently. Another motivation is to provide an approach for developing innovative applications quickly by composing existing executables and their local platforms as required by modern scientific problems. Based on Grid computing and software component technology, we have devised a component model AOD for on-demand computing on the computational grid.

The AOD provides a mechanism for encapsulating legacy applications as autonomic components, which are called grid-programming components (GP components). Every GP component encapsulates a collection of resources independently, and abstracts their computability to be a set of high-level and domain-termed operators on its interface. Each operator specifies some relatively complex computation that the GP component can implement with these encapsulated resources. In grid applications, these operators are referred to specify the computation that a grid application requires the GP component to complete. And these operators hide the required computation's implementing details from grid applications. A grid application is a composition of references to operators provided by GP components, and is represented as an acyclic and directed graph. Every node in the graph is a reference to an operator of some GP component and specifies a piece of computation required by the application. The graph sets a workflow for these computation pieces with its edges. When a grid application is running on the computational grid, the referred GP components are invoked concurrently and dynamically by the AOD, and each GP component is assigned a piece of computation. The task of coordinating different GP components is also left to the AOD.

In the next section, we first detail the concept of *grid-programming component*. The AOD is introduced in section 3. Section 4 presents a prototype of AOD and experimental results. Related works are overviewed in section 5, followed by a conclusion of this paper.

2. GRID-PROGRAMMING COMPONENT

We define a *grid-programming component* (GP component) to be an extensible entity associated with some domain concept, which encapsulates a collection of legacy applications and provides a set of data-processing functions for developing grid applications. These legacy applications and their target platforms constitute its underlying resources,

* This work was supported by National Natural Science Foundation of China (No. 60303001, No.60173004).

and its interface is a set of high-level operators. Every operator denotes one function in domain terms, and is automatically implemented on the computational grid. The underlying resources are augmented with necessary domain-specific expertise, so that the grid-programming component can schedule them to implement its operators efficiently and dependably. Different encapsulated applications are allowed to differ in functionality, performance, syntax and semantic of arguments, programming languages and target local platforms, etc.

To input and output arguments for the computation implied by an operator, a GP component provides a set of IO ports. All arguments required by an operator are input and output through a combination of IO ports, and each argument is transferred in data files through one IO port. An IO port is either an IN port for inputting or an OUT port for outputting some argument, and is associated with a *file-composing-descriptor* that provides a high-level description for the transferred argument in domain terms. A *file-composing-descriptor* (FC descriptor) is a file structure description for some kind of complex data objects, consisting of a set of file names and their semantic interpretation. It specifies a mapping from offsets of these files to elements of the corresponding data object type. Generally, a FC descriptor is independent of any GP component and defined by domain experts. The FC descriptor associated with an IO port decomposes every transferred argument into several components, and each component is identified by some specific file name contained in the FC descriptor. For convenience, we denote an argument component with the expression *port.file-name*, where *port* is the IO port that is responsible for transferring the argument, and *file-name* is the file name specified by *port*'s FC descriptor to identify the argument component.

With the concepts of file-descriptor and IO port, a job submitted to a GP component can be described formally. Assuming *gpc* is a GP component, *task* is a job submitted to *gpc*, and *op* is the operator that implies the computation desired by *task*. Then *task* can be described with a list of triplets; and each triplet is in the form of $\langle port.arg, file, loc \rangle$, where *port* is an IO port employed by *op* for transferring some argument, *arg* is a file name occurred in *port*'s FC descriptor, *file* is a data file at the URL location *loc*. The triplet specifies that *file* is provided as the argument component denoted with *port.arg* when *task* is submitted to *gpc*. The list contains exactly one triplet for every argument component that is transferred through some IN port employed by *op*. For every argument component of an operator, there is exactly one relevant triplet in the list if it is transferred through some IN port; otherwise, there can be any number of relevant triplets in the list.

For every job submitted to a GP component, the GP component is responsible for all of its implementation details on the computational grid. With the augmented expertise, the GP component first selects a subset of its underlying resources for the submitted job, according to the job's resource requirements and dynamic statuses of these underlying resources. Then resources that are essential to the job are reserved, and the GP component returns the submitter a handle to notify the success of the submission. Temporary storage for arguments is an example of these essential resources. The returned handle contains information about the reserved resources. Finally, the GP component begins the job's

implementation by fetching the job's input argument components from their original URL locations. With the augmented expertise, the selected resources are scheduled and coordinated automatically to complete the job. The GP component will send job's submitter a SUCCESS event after it has transferred every output argument component to the argument's destination specified by an URL location. And before the submitter has freed the job's handle, the job's result will be automatically buffered in some temporary storage and can be retrieved through the handle.

2.1 GP Component's framework

Every GP component details its underlying resources and the augmented domain-specific expertise in a configuration descriptor, as illustrated in Figure 1. The configuration descriptor begins by declaring a name for identifying it on the computational grid. The name usually consists of terms focusing on a concept of some problem domain, serving as an alias for referring the GP component in grid applications. Next is a list of IO ports on the GP component's interface. Then operators provided by the GP component are independently declared in detail. Every operator's declaration consists of *operator-name*, *IN-port-list*, *OUT-port-list* and *operator-body*. *operator-name* is its alias in domain terms. In grid applications, the alias is referred together with the GP component's name to indicate the operator identically. IN ports and OUT ports employed by the operator are listed in *IN-port-list* and *OUT-port-list* respectively. Every listed IO port is responsible for transferring one argument. *operator-body* provides the detailed domain-specific expertise that is required to implement the operator. The GP component is registered to the AOD by submitting a configuration descriptor, which is used by the AOD to configure and create it on the computational grid.

```

grid-programming component name
IO-port-list
operator-declaration
    operator-name1
        IN-port-list
        OUT-port-list
        operator-body
    operator-name2
        IN-port-list
        OUT-port-list
        operator-body
    .....

```

Figure 1 GP component's configuration descriptor

A GP component allows each operator to have more than one implementation, and every candidate implementation is independently provided by some local platform. Different implementations may differ in efficiency and resolvable problem size. The operator's *operator-body* details not only the policies and mechanisms for dynamically selecting a local platform for every submitted job, but also the information required to complete the desired computation on any selected local platform. Three kinds of domain expertise are provided. The first kind of expertise is about analyzing a job's arguments dynamically, so as to get the job's resource requirements. Another kind of expertise is about selecting a local platform for every submitted job, according to the job's resource requirements and the dynamic status of every candidate local platform. The last kind of expertise is about allocating and

scheduling resources to complete jobs on any candidate local platform.

The structure of an *operator-body* is illustrated in Figure 2. It begins with declaring a set of inspectors for querying information about a job's input arguments and the status of every candidate local platform. Examples of information queried by an inspector include sizes of a job's input arguments and available storage on some specific local platform, etc. Next is the declaration of a set of analyzers for deducing a job's resource requirements from information queried by the operators declared above. Behind the analyzers, an evaluator is declared. It synthesizes information about a job and a candidate local platform's status, so as to evaluate the job's efficiency on the local platform. The inspectors for querying information about a job's input arguments, the analyzers and the evaluator are independent executables. Before a GP component is created on the computational grid, these executables should have been installed on some default host that's specified by the AOD for managing all created GP components. For an inspector that queries information about statuses of candidate local platforms, every candidate local platform independently provides an implementation that is an executable too. To get the required information, an inspector may involve some simple processing about a few elements contained in a job's input arguments.

```

inspector-declaration
analyzer-declaration
evaluator

Candidate-Description
  candidate1
    local-platform
    criterion-range-list
    resource-booker
    module-list
    dependency-description
  candidate2
    local-platform
    criterion-range-list
    resource-booker
    module-list
    dependency-description
  .....

```

Figure 2 Structure of an operator-body

The *operator-body* structure ends with a set of *candidates*, and every *candidate* independently details the information of one operator implementation. A *candidate* consists of *local-platform*, *criterion-range-list*, *resource-booker*, *module-list* and *dependency-description*. *local-platform* declares a local platform that provides one candidate implementation for the operator, including its host identifier and login information. *criterion-range-list* sets a return-value range for every analyzer declared in *analyzer-declaration*. If any analyzer's return value for some job is out of the relevant range, then *local-platform* cannot meet the job's resource requirements. Both *resource-booker* and modules declared in *module-list* are independent executables installed on *local-platform* in advance. *resource-booker* is responsible for allocating resources that are essential for completing a job on *local-platform*, such as reserving storage and processors, etc. Every module's

declaration specifies both its name and usage on *local-platform*. Some modules may be legacy applications that have existed before. These modules play two roles. One is to perform some piece of data processing for the job. Another is to do some kind of data pre- or post- processing, such as preparing some module's arguments from the job's arguments and mid results returned by other modules, translating some module's result into the job's arguments etc. *dependency-description* details the dependency between different modules.

For every job submitted to a GP component, the GP component first selects a proper local platform for it. All inspectors and analyzers declared in the corresponding *operator-body* are executed one by one, so as to get information about the job's resource requirements and current status of every candidate local platform. According to the results of these analyzers and the criterion ranges specified by every *candidate* in the *operator-body*, the GP component judge whether a *candidate* is applicable to the job and invokes the *operator-body*'s *evaluator* to perform an efficiency evaluation for each applicable *candidate*. The *candidate* with the best evaluation result is selected for the job. With the support of AOD, the GP component invokes the selected *candidate*'s *resource-booker* remotely. After all essential resources have been successfully reserved by the *resource-booker*, the GP component returns the job's submitter a handle, and creates a schema for completing the job on the selected *candidate*'s *local platform*. The schema specifies all details for completing the job on the *local platform*, including the URL location of every argument component, information about the resources reserved by *resource-booker*, and the executing order of the modules declared in the selected *candidate*'s *module-list*, etc. Finally, with the support of AOD, the schema is carried out and the job is completed on the *local-platform*.

2.2 Developing Grid-Programming Components

As discussed above, a GP component encapsulates a set of executables, and these executables should have been installed on distributed local platforms before it is created on the computational grid. When it is registered to the AOD, its configuration descriptor is also required. Therefore, developing GP components requires specialized knowledge in programming languages, computer architectures, system administration, computing complexity analysis and problem domains.

Although a GP component's development requires the collaboration of domain experts, system administrators and programmers, different developers can contribute to it independently. Its configuration descriptor provides a high-level description for its structure, and explicitly specifies the syntax and semantic of these encapsulated executables in domain terms. It is independent of the development and installation of any encapsulated executable. With this description, different executables can be concurrently developed. At the same time, every operator is independent to each other, and its candidate implementations are independent too. Therefore, the development of a GP component can be divided into several independent tasks. Each task can be taken on by one group to provide the policies, mechanism and information for one operator. Furthermore, every task can be divided into a set of sub-tasks, and each sub-task is responsible for one candidate implementation that is independent too.

Two other strategies are employed to simplify the development of a grid-programming component. One is to reuse the legacy applications that have existed before for local platforms such as clusters, MPPs and workstations. Every legacy application can be directly encapsulated into a GP component as one of its modules. The framework illustrated in Figure 1 also allows a legacy application to be encapsulated by multiple GP components. Another is the extensibility of a GP component, and there are three approaches to extend or modify an existent GP component. One approach is to add new IO ports and operators by extending the configuration descriptor directly. The second approach is to modify the GP component's configuration descriptor by adding new *candidates* or deleting old *candidates*, so as to optimize relevant operator's performance. The third approach is to modify the encapsulated modules directly and has no affect on the GP component's configuration descriptor.

3. AN ARCHITECTURE FOR ON-DEMAND COMPUTING

In this section, we introduce the mechanism employed by our AOD to support on-demand computing on a computational grid. To solve a complex and large-scale scientific problems on the computation grid, multiple GP components are employed by the grid application, and each one is responsible for a piece of computation. These employed GP components are invoked and coordinated by the AOD to complete the desired computation automatically at runtime.

3.1 Developing Grid Applications With GP Components

In the AOD, a complex problem is divided into several concurrent and relatively simple sub-problems. The grid application represents every sub-problem with a reference to some proper GP component operator. These references are connected with directed edges to specify the problem domain's concurrency. When the application is submitted to run on the computational grid, the AOD will automatically create a formal sub-problem description for every reference, according to the application's arguments and the connected edges. The referred GP components are invoked concurrently, and each is provided a formal description for the corresponding sub-problem. The AOD is also responsible for transferring data objects and communicating messages for the invoked GP components on the computational grid. Since most scientific problems are very time-consuming, we assume that the cost for transferring a data object is much less than that required by a sub-problem.

Every directed edge in a grid application connects an IN port of some reference ref_1 with an OUT port of another reference ref_2 , specifying a data object exchanging between ref_1 and ref_2 . When the application is running, ref_2 's result specified by the connected OUT port will be transferred to ref_1 as its argument specified with the connected IN port. We refer the sub-problem specified by ref_2 as a precedent of the sub-problem specified by ref_1 ; and the sub-problem specified by ref_1 is referred as a successor of the sub-problem specified by ref_2 . For a reference, a data object should be provided for every argument that is specified with some IN port, and the data object can be an argument of the application or be specified with a directed edge. One OUT port of a reference is allowed to be associated with any number of directed edges, and it can also specify an output argument of the application. Every application argument is represented by a collection of

data files that are specified by their file names and URL locations, and every data file represents one argument component.

3.2 Architecture of AOD

The AOD provides an environment for developing and running grid applications. Its architecture is illustrated in Figure 3, consisting of **Repository**, **Scheduler** and **Broker**. The repository provides support for both developing GP components with legacy applications and developing grid applications by composing GP components. With the support of Globus Toolkit [3], combination of the scheduler and the broker provides the running environment for grid applications. The AOD requires that both the repository and the scheduler should be specified a host respectively. On every host that provides underlying resources for GP components, there exists an instance of the broker.

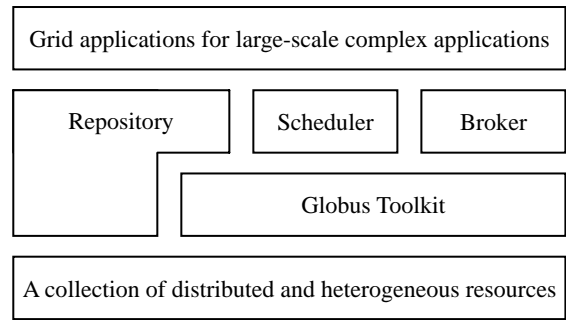


Figure 3 Illustration of AOD

The repository is responsible for configuring every registered GP component on the computational grid, and provides a universal environment for every configured GP component to select and schedule its underlying resources. This environment is independent of the GP component's candidate local platforms. When a GP component is registered, its configuration descriptor is submitted to the repository. With the configuration descriptor, the repository then configures the GP component on the computational grid. After a GP component has been configured, its configuration descriptor is reserved by the repository, and can be retrieved to get the GP component's interface or be replaced by a new one to extend the GP component. An existent GP component is automatically reconfigured if a new configuration descriptor is submitted to replace the old one.

On every candidate local platform of some GP component, the GP component's behaviors are conducted by a local instance of the broker. It provides three kinds of services for both the scheduler and GP components. The first kind of services is to start up an executable for a GP component on the local platform and send its result to the GP component after the execution. GP components use them to invoke the inspectors and resource-bookers declared in their configuration descriptors. If the invoked executable is an inspector, the broker then sends its result to the invoker directly. When a resource-booker is invoked, it creates a handle to identify both the local platform and the resources reserved by the resource-booker, and returns the handle to the invoker. The second kind of services is to schedule a list of executables and system shells to run locally, according to a prepared schema. They are used by a GP component to complete some submitted computation with the resources identified by a valid handle, after the GP component has got the handle and has

created a schema for the computation. The broker is also responsible for monitoring the computation's process. For every failure that has occurred and stopped the computation, the broker creates an ERROR event and sends it to its subscribers, so as to notify the failure immediately. After the computation has completed, a SUCCESS event is created and sent to its subscribers. The last kind of services is for subscribing events related to some handle that has been created by the broker instance itself. They allow both GP components and the scheduler to subscribe for events related to handles that have been created locally.

The scheduler provides an identical and universal entry for users to run grid applications on the computational grid. For a grid application submitted to the scheduler, it dynamically invokes and coordinates these referred GP components, so as to complete and optimize the computation. For every reference *ref*, the sub-job represented by *ref* is submitted to the GP component specified by *ref*, and different GP components are invoked concurrently. A sub-job is ready to be submitted only after all of its precedents have been completed. At first, only the sub-jobs who have no precedents can be submitted. When a sub-job is ready to be submitted, the scheduler first creates a formal and detailed description for it, according to the information provided by the application, the information attached to handles of its precedents and the application's arguments. Except the output arguments that are associated with some edges, every argument of the sub-job is provided a value in the description. Next, the scheduler submits the description to the corresponding GP component. The submission is completed when a handle is returned from the GP component, and the scheduler then subscribes for events related to the handle. Afterwards, it begins to submit any other un-submitted sub-job if all of its precedents have been completed. When the SUCCESS event of some submitted sub-job has been received, its successors are checked one by one to judge whether they can be submitted. Computation of the whole job is completed after every sub-job's SUCCESS event has been received.

4. IMPLEMENTATION AND EXPERIMENT

We have implemented a prototype for the AOD, as illustrated in Figure 4. In this prototype, real-time messages are exchanged on the computational grid when a grid application is running, so as to coordinate distributed resources dynamically. The scheduler, the repository and every broker instance specify a local TCP/IP port respectively for receiving messages from the computational grid. In order to improve the availability of shared resources, a GP component keeps inactive when no grid applications require it to perform any computation. When some piece of computation is submitted to an inactive GP component, the repository will autonomously activate it with the activator.

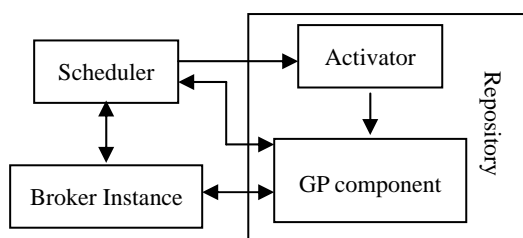


Figure 4 Implementation of AOD

When a GP component is scheduled to perform some piece of computation for a grid application, the scheduler submits the computation by sending the repository a message, which contains both the GP component's name and the computation's description. The message is delivered to the activator directly, and the activator extracts out the GP component's name and the computation's description. Then the activator invokes the GP component if it is inactive and passes it the description. Next, the GP component sends messages to the broker instances distributed on the computational grid, in order to invoke the relevant inspectors remotely and get the dynamic status information of candidate resources. The results of these invoked inspectors are sent to the repository concurrently, and every message is dispatched to the corresponding invoker. Once the GP component selects a local platform for the computation, it sends the local broker instance a message for reserving necessary resources. Finally, the GP component sends the computation's description to broker instance of the selected local platform. The selected broker instance then exchanges real-time messages with both the GP component and the scheduler respectively, so as to notify the occurred failure and whether the computation has completed successfully.

This prototype provides three XML schemas for programmers. The first schema is for domain experts to define FC descriptors. The second one is for developing the configuration descriptors of GP components, and the last one is for developing grid applications. We also have developed a tool for running grid applications with Internet browsers. The broker's instances on every local platform are managed with GRAM, and messages on the computational grid are exchanged with Globus IO. During a GP component is performing some submitted computation, arguments of the computation are transferred with GridFTP.

4.1 Experiment and Evaluation

In the rest of this section, we present a demonstrative example and its experimental result to evaluate both the AOD and the prototype. The example is some kind of simplified pre-stack migration in oil-prospecting data processing, and its computation can be divided into four pieces: a pre-processing operation, a voice-filtering operation, a Q-compensating operation and a synthetic operation. The pre-processing operation is to delete the invalid elements of some primal sampling data, and its results are passed to the voice-filtering operation and the Q-compensating operation respectively. The synthetic operation creates the final result by synthesizing the results of the voice-filtering operation and the Q-compensating operation.

Each of the operations has been independently implemented by an existent application, and these four applications are different in the syntax and semantic of arguments, target platforms, programming languages and models. The applications for the pre-processing operation and the voice-filtering operation are parallel, and they were developed with HPF. The other two applications are serial, since both the Q-compensating operation and the synthetic operation involve irregular computation. One was developed with C, and the other was developed with F77. We developed four GP components *prePrc*, *voiFilt*, *qCom* and *Synth*. Every GP component provides one operator on its interface, and encapsulates one legacy application and several additional executables. The additional executables were specially developed, and their function is to perform data transformation

between arguments of the GP component's operator and that of the application. The original sampling data of our experiment is about 2 GB, consisting of two binary data files. All of the legacy applications and the original sampling data are from the Computer Center of Shengli Oil Field.

The evaluation was performed on a test-bed that consists of five workstations and a Beowulf. All of the six hosts are connected to the college network of Peking University. The two HPF applications were installed on the Beowulf, and each serial application was installed on a different workstation. The original sampling data and the final result were stored on the third workstation. The rest two workstations run the scheduler and the repository respectively. Table 1 is the experimental result. The result shows that the distributed resources were dynamically scheduled to complete the demonstrative example's computation collaboratively and concurrently. Although additional executables were developed to implement the interoperation between different legacy applications, the programming task was very simple comparing with the complexity of any of the legacy applications. Furthermore, every GP component is universally sharable on the computational grid. When the grid application was running, different GP components were scheduled to complete independent pieces of computation with distributed resources. Therefore, its efficiency and resolvable problem size go beyond the computability of any single host or legacy application.

Table 1 Experimental Result of a Demonstrative Example

	Computing host	Working directory	Start time	End time
<i>preProc</i>	162.105.203.100	/home/chen/lyan/test1/	16:43:41 10/3/03	16:45:39 10/3/03
<i>voiFilt</i>	162.105.203.100	/home/chen/lyan/part1/	16:45:47 10/3/03	16:46:58 10/3/03
<i>qCom</i>	162.105.203.38	/home/aitest/oil/part2/	16:45:47 10/3/03	16:48:16 10/3/03
<i>Synth</i>	162.105.80.17	/home/globus/lyan/test2/	16:48:31 10/3/03	16:55:26 10/3/03

5. RELATED WORKS AND CONCLUSION

In recent years, the challenge of developing grid applications has been investigated extensively. The Open Grid Services Architecture (OGSA) [2] is the first effort to standardize Grid functionality and produce a Grid programming model consistent with trends in the commercial sector. It integrates Grid and Web services concepts and technologies. In this architecture, heterogeneous and distributed resources are encapsulated to be grid services with standard interfaces and behaviors. Every grid service is implemented independently on some local platform. However, in the first version for OGSA, all grid services are conformed to the Open Grid Service Infrastructure (OGSI) specification. One limitation is that there is no uniform way to schedule competing grid services universally for load balance and reliability of grid computing. Another limitation is that OGSI does not provide the mechanism for communicating messages on the computational grid, and this mechanism is necessary for coordinating complementing grid services to perform large-scale and complex scientific computation. The OGSI/WS-RF [4] has been proposed to replace OGSI in the early of this year.

XCAT [5,6] is an attempt to build an application component framework on top of OGSA. It provides an approach to

compose grid services in grid applications, hence to support distributed computation on the computational. Its current implementation neither takes account into the fact that many competing services proliferate on the computational grid. Like XCAT, ICENI[7] is also component-based and support distributed computation on the computational grid. Different from XCAT, ICENI seeks to annotate the programmatic interfaces of grid services using WEB Ontology Language, and allows syntactically different but semantically equivalent services to be autonomously adapted and substituted.

Comparing with XCAT and ICENI, AOD provides not only the mechanism for scheduling competing resources universally, but also the mechanism for querying the dynamic statuses of these resources. It allows domain experts and administrators to customize its resource selecting policies with domain-specific expertise. With these mechanism and policies, AOD allows its GP components to be adapted to both the resources requirements of grid applications and the dynamic statuses of competing resources, so as to utilize all available resources of the computation grid to provide on-demand computing services. We are going to substitute grid services for the candidates of GP component operators in AOD, in order to be more compatible with OGSI.

6. REFERENCES

- [1] I. Foster, C. Kesselman, S. Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001.
- [2] I. Foster, C. Kesselman, J. Nick, S. Tuecke. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
- [3] Globus Toolkit. <http://www.globus.org/toolkit/default.asp>.
- [4] WS-Resource Framework. <http://www.globus.org/wsrf>.
- [5] Dennis Gannon, Sriram Krishnan, Liang Fang, Gopi Kandaswamy, Yogesh Simmhan, and Aleksander Slominski. On Building Parallel and Grid Applications: Component Technology and Distributed Services. <http://extreme.indiana.edu/labpubs.html>.
- [6] Sriram Krishnan and Dennis Gannon. XCAT3: A Framework for CCA Components as OGSA Services. In Accepted for publication to HIPS 2004, 9th International Workshop on High-Level Parallel Programming Models and Supportive Environments. IEEE Computer Society Press, 2004. <http://extreme.indiana.edu/labpubs.html>.
- [7] J. Hau, W. Lee, and Steven Newhouse. Autonomic Service Adaptation using Ontological Annotation. In 4th International Workshop on Grid Computing, Grid 2003, Phoenix, USA, Nov. 2003.

Yu Huashan is an assist professor in the School of Electronics Engineering and Computer Science, Peking University. He received his Ph.D. in computer science from Peking University in 2001. His research interests include parallel computing and Grid programming for large-scale scientific computation.

Xu Zhuoqun is a senior professor and doctoral supervisor in the School of Electronics Engineering and Computer Science, Peking University. His current research areas include parallel computing, geography information system and artificial intelligence.

Researches of Key Technologies for Data Grid*

Fansong Meng, Zude Zhou, Quan Liu
 Information Engineering School, Wuhan University of Technology
 Wuhan, Hubei 430070, China
 Email: xiaomeng21c@mail.china.com Tel: 027-50105616, 13016482446

ABSTRACT

Data Grid is a large-scale, scalable framework structure of sharing and managing storage resources and distributed data resources in the Grid environment. It adapts to demands of data sharing and handling that are needed by data intensive applications also, it provides users with transparent mechanisms of accessing heterogeneous data resources. After analyzing the characteristics and architecture of Data Grid, this paper researches key technologies of Data Grid, such as security and architecture designs etc. In the end, it gives some Data Grid research programs.

Keywords: Grid, Data Grid, architecture, data access, GridFTP, replica management

1. INTRODUCTION

In an increasing number of scientific disciplines, large data collections are emerging as important community resources. In domains as diverse as global climate change, high energy physics, and computational genomics, the volume of interesting data is measured in terabytes and petabytes, the communities of researchers that need to access and analyze this data (often using sophisticated and computationally expensive techniques) are often large and are almost always geographically distributed, as are the computing and storage resources that these communities rely upon to store and analyze their data.

This combination of large data set size, geographic distribution of users and resources, and computationally intensive analysis results in complex and stringent performance demands that are not satisfied by any existing data management infrastructure. A large scientific collaboration may generate many queries, each involving access to the supercomputer-class computations on-gigabytes or terabytes of data. Efficient and reliable execution of these queries may require careful management. How to store, distribute, organize and manage, handle and dig gigantic distributed data have become a chief problem. Development of Data Grid technology^[1] offers a effective technological path for solving this problem. Through developing many resources like data that are distributed on network, Data Grid can form a single virtual data access, management and handling environment, help users shield low-level heterogeneous physical resources and constitute data access, storage, transmission, management and service architecture of gigantic distributed data.

2. CHARACTERISTICS

AND

ARCHITECTURE OF DATA GRID

Data Grid originates from Grid^[2]. It's applications and implementations of network technologies in data management environment, also it's a new architecture of transparently accessing heterogeneous data resources for constituting Grid environment. Data Grid has typical characteristics and layering architecture.

2.1. Characteristics of Data Grid

From the perspective of architecture, Data Grid has the following features:

- ◆ Heterogeneity. Data Grid contains many kinds of heterogeneous data resources. Based on many levels such as architecture, data access methods and API(Application Programming Interface), Different data resource has different constitution.
- ◆ Scalability. Data Grid's scale can be changed from local area system containing few data resources to cross-continent wide area data Grid consisting tens of thousands of data resources. But meanwhile, a problem arises: the increase of Grid resources and further geographical distribution will lead to the decrease of the performance and network transmission delay etc, Data Grid has to adapt to such changes.
- ◆ Adaptability. There are many data and storage resources in wide area systems and these resources have very high failure probabilities. Therefore, Data Grid should help users and applications shield such failures and dynamically adapt to these situations. In addition, Data Grid's resources usually vary because of their geographic distribution and system complexity, Data Grid should also adapt to such unpredictable structure.
- ◆ Multi-level management domains. Because usually resources and storage systems that comprise Data Grid belong to diverse institutions and organizations and use different security mechanisms, corporate participations of all institutions and organizations are essential to solve problems concerning multi-level management domains.

From the perspectives of designs, Data Grid has such characteristics:

- ◆ Low-level structure-independent. Data Grid is independent of low-level structure, such as computer hardware, OS and storage systems.
- ◆ Implementation methods-independent. Data Grid allows users to customize or modify function implementation methods while don't change it's systematic structure.
- ◆ Computational Grid-compatible. Data Grid can use protocols, authorization and resources management Computational Grid directly.
- ◆ Architecture-unified. Protocols and Interfaces of different Data Grids have to be unified to realize the aim of interoperability.

*National Key Scientific and Technological Project(2003BA103C)

2.2. Architecture of Data Grid

Architecture of Data Grid can be depicted by the well-known five layer funnel structure^[3]. As illustrated in Figure-1, located in the lowest layer is the *fabric*, it faces all kinds of concrete physical (or logical) resources. Through managing these local resources, fabric offers management of these resources and control interfaces for upper layer. *Connectivity* mainly provides the lower layer's physical resources with secure communication capacity which is the premise of interoperability between resources and makes single isolated resource set up connection. *Resource* reflects characteristics of abstract local resources while its upper layer *collective* collects the lower layer's single resource to harmoniously solve problems among resources. *Application* cares about what kind of resources can be provided by lower layers to virtual organizations and solve concrete problems of different virtual organizations.

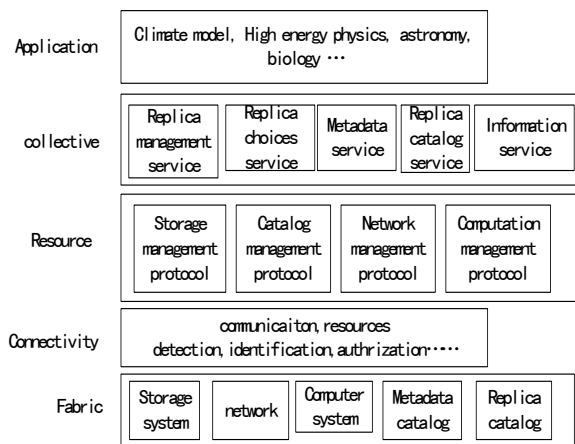


Figure 1 Architecture of Data Grid

3. ANALYSIS AND RESEARCH OF KEY TECHNOLOGIES CONSTITUTING DATA GRID

Based on fundamental functions, Data Grid extends data management functions and provides all kinds of services of information-related. It needs operating gigantic data and have such characteristics as geographic distribution、environment heterogeneity、individual autonomy、resource dynamics and interoperability. These characteristics determine that constituting Data Grid is relatively hard. Therefore, the following key technologies should be solved properly.

3.1. Data access and Metadata Access

Data access and metadata access are two fundamental services of Data Grid. Data are stored in storage systems. They are handled and analyzed by programs. While metadata are related to data, such as size of data、location of data etc. Data access service provides mechanisms for access、management and third-party transmitting data. While Metadata access service provides mechanisms for accessing and issuing metadata etc. To discuss their conceptions separately is to offer flexibility for the implementation of Data Grid. While in realistic implementation, they can be combined with one another.

3.1.1. Data access service

Storage systems are fundamental components of Data Grid, it realizes functions of setting up、deleting、reading、writing and controlling files. Its attributes include titles、storage capacities and access limitation conditions etc.

Usually, users access storage systems and files through API. Here the API is the extension of traditional conception, such as, it can support remote reading and writing files etc. During the implementation process, storage systems should take other factors into considerations, for example, access functions have to be combined with security strategies; storage systems should examine and determine its performances and provide them to users so that users can optimize access strategies; applications should provide storage systems with access modes etc. In addition, storage systems should take fault-tolerance and robustness etc into account when design systems.

3.1.2. Metadata access service

Metadata can be divided into the following categories according to its depicted content:

- ◆ Application metadata. It depicts files' content or some information that are useful for handling the applications of these files.
- ◆ Replica management metadata. It is responsible for replica management of data objects, including information of mapping of files into storage systems positions etc.
- ◆ System configuration metadata. It depicts the structures of Data Grid itself, such as, internetworking、details of storage systems(capacity、strategy etc).

Every metadata has its own features in such fields as application scopes、upgrading mechanisms、logical relationship with other Grid components. To issue and access different kinds of metadata, it's necessary for metadata to offer consistent application methods、single application interfaces etc.

Usually, metadata access service is designed into a distributed service with layering architecture. Such architecture has many advantages: offering scalability and avoiding single failure point. Efficiency of such architecture can be amended by making the most of layering attributes of metadata service itself.

3.2. Data Replica Management Service

Realizing the functions of data replica management in Data Grid is to acquire better data access efficiency and fault-tolerance. During the process of data-intensively large-scale distributed collaborative applications, on one hand, the users-groups of data are geographically widely distributed; on the other hand, data are also distributed and stored on different places. As a consequence, maybe computation on a certain place needs data of another place, so problem of access time occurs. To reduce access time through networks, some data stored on remote machines can be copied on local machines or store some data on different machines, when such data are needed, computation nodes can access the data form the nearest node. Both the methods lead to several copies of a certain data in the system. Data replica management service includes:

- ◆ To generate the copies of all of a data packet or some data.
- ◆ To register new copies into replica file directories.
- ◆ Users or applications inquire replica file directories to acquire all physical copies of a certain file or files.
- ◆ To choose the most suitable replica file for users or applications to access according to information provided by Grid information service and based on storage and Grid performance predictions.

3.3. Data Transmission Protocol GridFTP

Currently, many applications use different storage systems, but usually these systems are not compatible with each other, because they use different software and protocols. Consequently, problems such as security and high-speed of data transmission occur. To solve these problems, Globus^[5] advanced GridFTP mechanism which is based on FTP, and comprehensively extend its functions. In addition to inheriting FTP's features as good scalability, wide application, standardization etc, GridFTP also has many new characteristics that adapt to Grid structures:

- ◆ Supporting GSI and Kerberos security mechanisms. GridFTP supports reliable and flexible security identification and integrity inspection. Also, users can control data integrity of GridFTP on different layers.
- ◆ Supporting the third-party controlled data transmission. To manage data groups of distributed communication systems, GridFTP has to offer the third-party controlled data transmission that is identified.
- ◆ Supporting parallel data transmission. In the WAN environment, GridFTP uses several parallel TCP flows to enhance the total bandwidth of data transmission and supports parallel data transmission and data channel extension.
- ◆ Supporting partial file transmission. Some applications may only need accessing parts of a certain remote file and it needs data transmission support, GridFTP supports data transmission from files' arbitrary places and effectively support partial file transmission.
- ◆ GridFTP can automatically adjust the size of TCP buffer/window, to utilize optimized TCP buffer/window can effectively enhance the functions of data transmission.
- ◆ Supporting reliable transmission and data resending. As far as many applications are concerned, they have to support reliable data transmission and fault-tolerant data transmission. GridFTP extends resending of failure data transmission and effectively support reliable transmission and data resending

3.4. Resource Scheduling Optimization and Remote Execution

In Data Grid, resources scheduling optimization and service execution is a key technological issue, it mainly includes request scheduling optimization, resources scheduling optimization and resource service execution. Request scheduling optimization should match users' resources requests and usable resources. When lots of users and applications appear simultaneously, we must optimize and plan multiple requested resources.

Remote execution service mechanism makes systems that are distributed on many places remotely activate & execute,

control, collect and find out the status information, also control the task execution processes of geographically distributed multiple systems.

3.5. Security

To deploy computation on WAN, assuring the security is vitally important. Grid security technologies provides basic security protection validation mechanisms to validate authorized users and resources, and it also provides interfaces for other security services to permit users to choose different security strategies, security levels, encryption methods and security facilities. It's demands and characteristics of Grid.

In the Data Grid environment, copy and buffering of data lead to security problems: a station buffers data located on another station. Because the two systems have different security protection mechanisms, measures and security levels, it's a very tough problem as to how to satisfy data owners' data protection security levels and strategies.

3.6. Architecture Design of Data Grid

Architecture design of Data Grid considers how to effectively organize all kinds of Data Grid services so as to constitute a high-efficient system and provides users or applications with what kind of method or interface. We have to take into account the relations between Data Grids when designing the architecture of Data Grid. Because to realize data management functions in the Data environment, Data Grid has to be founded on the basis of general Data architectures, that is, on one hand, to realize Data Grid's functions have to utilize other Grids' services (security service, resources scheduling service, performance service and information service etc.); on the other hand, the implementation of some Grids' services have to use functions provided by Data Grids' services.

Current research programs of Data Grid are all based on layering architecture. The lower layers mainly concern managing lower layers' resource and middleware and consider how to effectively realize functions of data access instead of a certain specific application strategy. For example, when realizing data moving functions, lower layers only consider how to move data with high speed and provide upper layers with some system interfaces, such as, fault handling interfaces etc. In a word, the layering architecture consists of a series of related, inter-independent or inter-dependent services, every service is responsible for realizing a specific function and it maybe dependent on other services.

4. RESEARCH EXAMPLES OF DATA GRID

Data Grid has comprehensive applications in such fields as biology, medicine, earth exploration, astronomy and weather etc. There are more and more Data Grid programs which can effectively help scientific research in these fields.

4.1. Globus Data Grid

Globus is most successful Grid research program. It develops a series of protocols, services, software libraries and toolkits to constitute a Grid environmental platform. Globus Data Grid can be divided into two layers: core service layer (CSL) and high-level service layer (HSL), in

which the HSL is founded on the CSL and uses services provided by the CSL. CSL provides general-purpose lower layer mechanisms and manages all kinds of storage systems so that high-level services and applications can access these systems through the same method.

4.2. Euro Data Grid^[8]

The final objective of Euro Data Grid is to develop a science and research(S&R) environment that adapts to next generation S&R. Researcher of this program think characteristics of next generation S&R work include: enhanced computation performance, handling and sharing large-scale data and wide area distributed scientific collaboration etc. These demands has appeared in many research work of scientific fields ,such as biology、physics and earth sciences etc. In these programs, all factors ,such as, resources' distributed characteristics、research groups' distributed attributes、gigantic capacities of databases and limited usable bandwidth etc have made resources-sharing become more complicated.

4.3. Grid Physics Network^[9]

Grid Physics Network (GriPhyN) is co-constituted by some experiment physicists and IT researchers. Its objective is to store and handle data containing millions of billions of bytes. Petascale Virtual Data Grids (PVDG)—core technology platform of GriPhyN—provides data handling-oriented computation platform for scientists throughout the world. Initial research works of GriPhyN include four application programs and two of them are CMS and ATLAS. Their purposes are to explore origins of substance and super-micro substance.

4.4. Earth system^[10] Data Grid

Earth system Data Grid is co-constituted by four DOE laboratories(ANL、LANL、LBNL and LLNL)、National Science Fund and two universities(University of Wisconsin and University of Southern California) and it's aim is to support high-speed access remote distributed large-scale climate model databases. It is founded on the basis of current technologies(DPSS、Globus)and develops a new "intelligent" middleware to realize goals of managing distributed data and transmitting high-performance data and remotely executing computers' components.

4.5. Other Data Grids

Particle Physics Data Grid^[11] (PPDG) started in 1999 and its purpose is to set up a Data Grid environment applied to high-energy physics and nuclear experiments. Current research program of PPDG is called SciDAC (Particle Physics Data Grid Collaboratory Pilot) which is a 3-years' research plan and will start a long physical experiment plan after 2006.

iVDGL^[12] is a global Data Grid connecting the US、Europe、Asia and South American, which is mainly applied to Physics and astronomy.

DataTAG^[13] is a large-scale continental Data Grid experiment platform, which has a close relationship with three Grid programs: GriPhyN、PPDG and iVDGL. On one hand, it provides a high-performance network connection from Geneva to Chicago(2.5Gbps), GriPhyN、PPDG and iVDGL all use it because it's a ideal choice of connecting the US and Europe; on the other hand, this program focuses

researching interoperability of Data Grids including DataGrid、CrossGrid and the three Grid program mentioned above.

5. CONCLUSION

Traditional centralized computation models are gradually developing into Grid computation models containing more computation and data resources. Data management and shared new architecture are gradually becoming hot topics of Grid researches. This paper depicts the fundamental characteristics and architecture of Data Grid and mainly discusses and researches key technologies of Data Grid. Undoubtedly, Data Grid is novel and people have different opinions about it and it should be further known、regulated、promoted and utilized. Of course, the researches and applications of a group of Data Grid programs provide directions for its researches and development in many fields and make it clear for light foreground of Data Grid. With the increasing maturation and comprehensive applications, Data Grid will surely further serve scientific engineering researches and production & applications experiments.

6. REFERENCES

- [1]. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of Network and Computer Applications*, 1999
- [2]. Ian Foster, Carl Kasselmann. *The Grid: Blueprint for a new Computing Infrastructure*. San Francisco, CA: Morgan Kaufmann, 1999
- [3]. I. Foster, C. Kesselman, S. Tuecke. *The Anatomy of the Grid: Enabling Scalable Virtual Organization*. *International J. Supercomputer Applications*. 2001,15(3). Also in: *Cluster Computing and the Grid*, 2001. *Proceedings. First IEEE/ACM International Symposium 2002*
- [4]. Heinz Stockinger, Asad Samar, Ian Foster. *File and Object Replication in Data Grids*. *Proc. Intl. Symp. On High Performance Distributed Computing*, IEEE Press, 2001
- [5]. <http://www.globus.org>
- [6]. W.Allcock, J. Bresnahan, I.Foster, L.Liming, J.Link, P. Plaszczac. *GridFTP Update*, Technical Report. January 2002.
<http://www.globus.org/datagrid/deliverables/GridFTPOverview2002.pdf>
- [7]. <http://www.npaci.edu/dice/srb>
- [8]. Euro Data Grid Project, <http://www.cern.ch/grid>
- [9]. Grid Physics Network Project. <http://www.gridphyn.org>
- [10]. Earth System Grid. <http://www.scd.ucar.edu.css.esg>
- [11]. <http://www.cacr.caltech.edu/ppdg/>
- [12]. <http://www.ivdgl.org>
- [13]. <http://datatag.web.cern.ch/datatag/>

Fansong Meng is a postgraduate of



Information Engineering School of Wuhan University of Technology. He graduated from Wuhan University of Technology in 2002 with specialty of Electronic and Information Engineering. His research interests are in network technologies and signal processing.



Zude ZHOU, professor, Ph.D supervisor, is the president of Wuhan University of Technology, his research interests are CNC theory and technology, intelligent control, digital manufacturing, reliability and fault diagnosis of the modern manufacturing systems etc.



Quan Liu is a Full Professor and the dean of Information Engineering School of Wuhan University of Technology. She is also a Ph.D supervisor. In recently years, she has published over 80 Journal papers. Her research interests are in computer network communication, signal processing and non-linear system theories and its

applications etc.

Global Grid Queue Services Architecture and Point-based Simulated Annealing Algorithm for Resource Scheduling

Shengjun Li^[1], Ruimin Shen^[1], Robert Lackman^[2]

1. Computer Science Department, Shanghai Jiaotong University, 200030, China

Shengjunli, rmshen@sjtu.edu.cn

2. Computer Science Department, China Ocean University

hubaoqingdao@yahoo.com

ABSTRACT

Grid resource management and scheduling strategy are key issues in grid service application and research. This paper proposes the overall global grid queue service architecture, which can provide standard API for Globus. Based on the system, a novel global scheduling algorithm: point-based simulated-annealing scheduling algorithm (P-SAS) is proposed. Compared with other scheduling algorithms such as random selection scheduling and best-random-n scheduling algorithm, P-SAS scheduling is shown to be the best in the grid simulation environment.

Keywords: grid queue, Globus, point-based, simulated annealing, scheduling, simulation

1. INTRODUCTION

Resource management and task scheduling are the key technologies in the grid service system. A huge amount of researches have been done on this issue in the conventional distributed systems and parallel computing. But, since the grid system has the characters such as geographical distribution, heterogeneous structure and intensive dynamicity [1], new algorithms are necessary to be researched in order to fulfill the needs of grid computing environment. Globus service [2] is the open standard architecture for the grid computing system in the world. All the higher-level grid application projects should be compatible to Globus. In this paper, one global queue service architecture that is compatible with globus, has been developed and the corresponding algorithm, point-based simulated annealing scheduling (P-SAS) is put forward. Compared with the other scheduling algorithms, P-SAS is shown to be the best in the simulated grid environment.

2. RELATED WORK

A lot of efforts have been taken to do research on the scheduling algorithm in the conventional distributed system and grid computing system. Paper [3] has suggested one heuristic scheduling algorithm, where the strategy is to select the minimum executing time to dispatch the task. Paper [4] proposed one genetic algorithm for the scheduling. Paper [5] aims at independent and equal-sized task. Paper [6] presents a meta-scheduler with a 2D chart and meta-scheduler types. But all the strategies only consider the executing time of the independent jobs and ignore the time to find the best matching resources. P-SAS overcome the shortcomings by taking the scheduling time into account and considering the multiple QoS demands and combined them to the measurement of points (point based).

3. SCHEDULING ARCHITECTURE

The scheduling service architecture has three layers as figure-1. The first layer is the job submission layer that is responsible for authentication and authority of submitted jobs. The second layer is the scheduling provider, which is in charge of putting jobs submitted into global queue, and treating them in FIFO strategy. The scheduler module performs the scheduling algorithm to select the best resources. Scheduling life cycle services module monitors the whole process of scheduling. The job information and resource specification will be modified to the uniform format such as XML document and then sent to the third layer—the resource discover and selection service module, which provides the information of the resources.

The scheduler and selection service modules provide standard API to access underline Globus architecture.

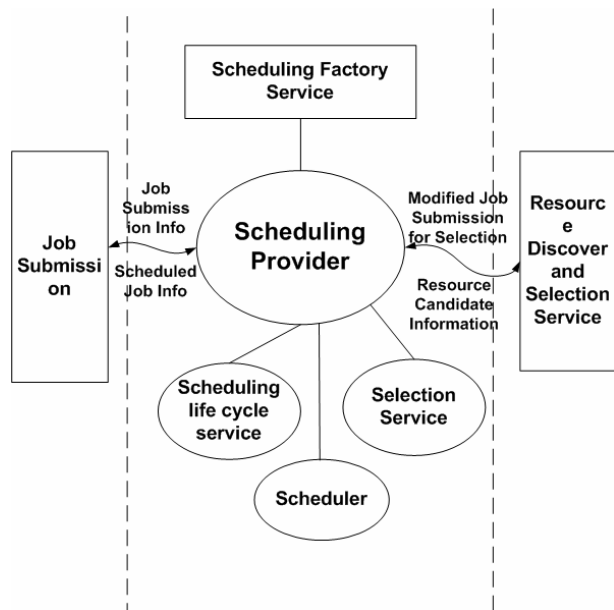


Figure 1

4. SCHEDULING ALGORITHM

To schedule the jobs efficiently, both the time for scheduling and the time for executing the jobs should be considered. Some algorithms are too complicated that the time for scheduling the algorithm is beyond the time for executing the job, and these kinds of algorithm should not be considered as good algorithm.

4.1 Random selection and best-random-n scheduling

The scheduling task is a complex one and takes a significant amount time. When the complexity of scheduling an application is sufficiently large, it may be the case that the scheduling phrase may be proportional to the executing phrase. The extreme case is that scheduling time may be longer than the executing time saved by using the scheduling algorithm.

Consider a random scheduling algorithm, which simply selects a resource at random without checking further. This scheduling algorithm is very fast and simple, so it will not produces an excessive delay of scheduling.

Best-random-n scheduling is an improvement of random selection algorithm. It randomly generates n schedulers and returns the one with the highest quality. This method maintains the speed of the random n algorithm while increases the chance to select high quality of resource.

4.2 Point-based simulated annealing algorithm

Paper [7] uses simulated annealing to address the issue of how many resources should be used when a problem can be split over a varying number of resources. We have shown that simulated annealing can be used to choose locations for component execution.

To schedule the resource more precisely, this paper considers the pointed based anneal simulation algorithm (P-SAS), Simulated Annealing is a generalization of the Monte Carlo method used for optimization of multi-variable problems. Simulated annealing is used to select how many resources in grid should be split over in order to achieve highest benefit.

The benefit of the resource in this paper is the point, which considers both computing power and network bandwidth. The formula for calculated the point is:

$$P = \sum_{i=1}^n RP_i - k \sum_{j=1}^m 1 / NP_j \quad (1)$$

Where P is the point, RP is the resource point, k is the constant weight, NP is the network point, n is the count of nodes. m is the count of connections between the nodes.

The resource point RP can be calculated as the following:

$$RP = CPU_speed / system.Load \quad (2)$$

The network point NP is the bandwidth between multiple nodes (Mbps).

Algorithm: point-based Simulated annealing

```

1: cSol ← generateNewSolution ()
2: cPoint ← getPoint (cSol)
3: While noAcceptedSolutions > 0 do
4:   noAcceptedSolutions = 0
5:   for l = 0 to MaxNoOfTrialSolutions do
6:     tSol ← generateTrialSolution ()
7:     tPoint ← getPoint (tSol)
8:     if acceptTrialSolution () then
9:       cSol ← tSol
10:      cPoint ← tPoint
11:      noAcceptedSolutions ++
12:   if noAcceptedSolutions >= maxAcceptedSolutions then
13:     break out of for loop
14:   end if
15: end if

```

```

16: end for
17: reduce T
18: end while

```

An initial schedule is generated at random and its point is calculated. A new schedule, a permutation of the previous is generated by moving one component (cpu, bandwidth, etc of node) onto a different resource is then created. The new schedule is either accepted or discarded as the new solution through the Metropolitan Algorithm. If the new solution has a higher point than the current point, it is accepted as the new selection. However, if the new solution has a lower point than the current solution, it is accepted with a probability by $e^{-d\beta/T}$, where $d\beta$ is the difference in point value between the two solutions, and T (temperature) is the control parameter.

Algorithm 2 Metropolis Algorithm

```

1: If  $d\beta < 0$  then
2:   return true
3: else if  $R < e^{-d\beta/T}$  then
4:   return true
5: else
6:   return false
7: end if

```

This process is repeated with each iteration consisting of either a maximum number of new solutions being accepted N , or a maximum number of solutions being considered M . At the end of each iteration T is decreased. Once iteration is completed with no new solution being accepted, the current solution is returned as the best available solution.

Low value of T decreases the probability that a solution with a lower point will be selected, as do large value of $d\beta$. At high values of T worse solutions are often accepted reducing the chance of the algorithm getting caught a local maxima. As T is decreased, so is the regularity with which worse schedules are accepted, allowing the algorithm to settle with the best algorithm found.

5. EXPERIMENTAL COMPARISON

To evaluate the effective of the algorithm presented, and select the best one to use, we run each scheduler on a range of different applications in a simulated grid environment. The simulated scheduling framework reads application and grid configuration data, which are put into files and answers queries from the scheduler with this information. This method allows the experiments being repeated run.

We build a simulated environment, which have 5 grid clusters, as the following table.

Table 1

Cluster	Num of nodes	CPU per node	CPU speed	Interconnect Speed
Cluster-1	8	1	2Ghz	100Mbit/s
Cluster-2	8	1	2Ghz	100Mbit/s
Cluster-3	4	1	2.4Ghz	1Gbit/s
Cluster-4	1	4	900Ghz	1Gbit/s
Cluster-5	1	8	750Mhz	5Gbit/s

Twenty-one applications each consisting of multiple components connected in a DAG configuration were tested. The DAG depth varies from two to seven. The number of the components in the application varies from two to thirty-seven, the point of each solution under consider are calculated by Eq.(1). The computational complexity of each component will between zero to ten minutes on a 2Ghz CPU, with an average of five minutes, and the communication between components with 100Mbit/s connection will take between zero and two minutes, with an average of one minute, both with Gaussian distribution.

To evaluate the algorithm, we examine the time between the submission and return of the application. Two distinct stages occur in this time period. First is the time of the scheduling time of the application, second is the time of the executing time of the application. Though the scheduling algorithms try to minimize the second stage, but at the cost of increasing the first stage. To evaluate the tradeoff, we consider the sum of the two stages to check the effective of the two algorithms.

Figure-2 and Figure-3 show how the three scheduling algorithms compared with under the condition of different components number and DAG depth.

These results are from the case that all the five clusters are available. The results show that the pointed-based simulated annealing produces the shortest time. The best-random-n algorithm's performance is the second and the random algorithm is the third.

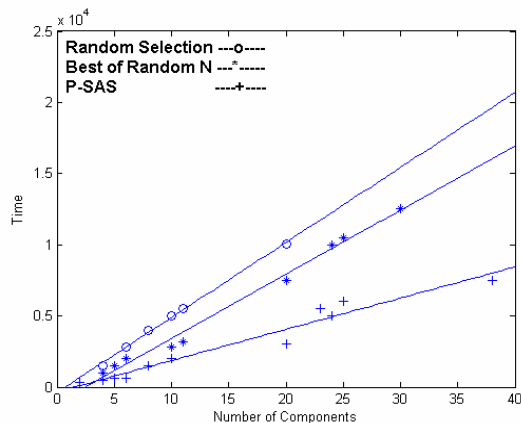


Figure-2

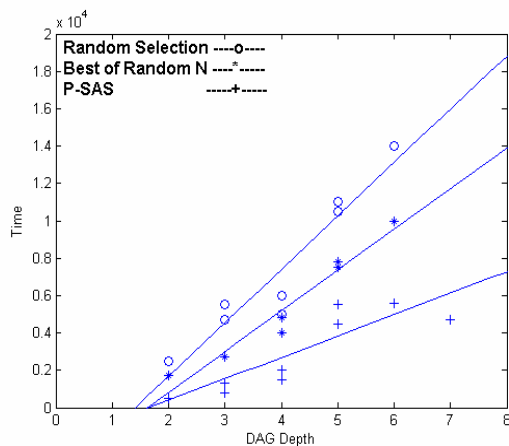


Figure-3

6. CONCLUSION

This paper presents the global queue scheduling architecture and corresponding algorithm—point-base simulated annealing scheduling (P-SAS). The architecture has three layers, which is compatible with globus service standard. Several algorithms are compared with P-SAS under the simulated Grid environment. Experiments results show that the P-SAS can produce better performance than random selection and best-random-n scheduling. Further research is to extend the P-SAS algorithm to the variant priority and real-time job scheduling.

7. REFERENCES

- [1]. I. Foster, C.Kesselman, Jeffrey, M.Nick: The physiology of the Grid: An Open Grid Services Architecture for Distributed system Integration. <http://www.gridforum.org/ogsi-wg/drafts/ogsa-draft2.9-2002-06-22.pdf>
- [2]. The Globus Project. <http://www.globus.org>
- [3]. A. Abraham, R.Buyya et.al: Nature's Heuristic for Scheduling Jobs on Computational Grids. Proc. of 8th International Conference on Advanced Computing and Communication, Cochin, India, 2000
- [4]. A.Y.Zomaya et.al: Observations on Using Genetic Algorithms for Dynamic Load-Balancing. IEEE Transactions on Parallel and Distributed Systems, 2001,9:899-911
- [5]. O.Beaumont and L.Carter: Bandwidth-Centric and Allocation of Independent Tasks on Heterogeneous Platforms. Proc. of the International Parallel and Distributed processing Symposium, 2002
- [6]. S.S.Vadhiyar and J.J.Dongara: A Met scheduler for the Grid. Proc of the 11th IEEE International Symposium on High Performance Distributed Computing.2002
- [7]. A.Yaikhani and Dongarra. Experiments with Scheduling: Using Simulated Annealing in a Grid Environment. In Grid, November, 2002



Shengjun Li received B.E. degree in Mechanics from Jilin University in 2000 and M.S. degree from Ocean University of China in 2003, respectively. He is currently a PHD student at computer science department of Shanghai Jiaotong University, China. His research interests include distributed system, performance analysis, grid computing and network QoS.

Ruimin Shen Received B.E. degree and M.S. degree both from Tsinghua University, China, in 1989 and 1992, respectively. He is currently a professor at computer science department of Shanghai Jiaotong University, China. His research interests include network multimedia, network QoS, Coding algorithm, and distance education.

The Application of Grid in Grid Services *

Zhao JianMin^(a), Zhu XinZhong^(b)

^(a) College of Computer Science and Engineering, Zhejiang Normal University

^(b) Institute of Computer Science Studies, Zhejiang Normal University

Jinhua, Zhejiang 321004, China

Email: znuzjm@mail.zjnu.net.cn _Tel.: (86) 05792282145

ABSTRACT

In recent years, the volume of datasets in modern large-scale scientific researches, information services and digital media applications is growing explosively, and the research about data grid technology becomes the new hotspot in the computer science all over the world.

Keywords: Grid, Globus, Grid Bank, Grid Market Directory, Grid services.

1. INTRODUCTION

Grid computing is gaining a lot of attention within the IT industry. Though it has been used within the academic and scientific community for some time, standards, enabling technologies, toolkits and products are becoming available that allow businesses to utilize and reap the advantages of grid computing.

As with many emerging technologies, you will find almost as many definitions of grid computing as the number of people you ask. However, one of the most used toolkits for creating and managing a grid environment is the Globus Toolkit.

And now building on both Grid and Web services technologies, people have complemented many Grid technologies such as Globus, Grid Bank and Grid Market Directory. And many Grid services appear and are changing our lives.

Therefore we will present most of our information and concepts within the context of the Grid.

So this paper is organized as follows. The conception of Grid and the difference between Grid and cluster computing, single system parallel systems and a Web service is presented in Section 2. Section 3 describes Grid technologies that are used in the current implementation. We conclude in Section 4 with a discussion of current system status and future work.

2. CONCEPTION OF GRID AND GRID COMPUTING

Let us start off with defining a Grid. However, it seems to us that various researchers have differing views on Grid computing mostly based on technologies or applications that they are developing and what they envision it to be. But I think the definition is as follows: "Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of services of heterogeneous resources distributed across "multiple" administrative domains based on their

availability, capability, performance, cost, and users' quality-of-service requirements". Like any distributed system, Grids need to address various issues and challenges including: security; autonomy; heterogeneity of resource access interfaces, policies, capability, pricing; data locality, dynamic variation in availability of resources, and complexity in creation of applications. Therefore, Grid follows a combination of hierarchical and decentralized architecture for resource management; and a layered architecture for implementation of various services.

The most common description of grid computing includes an analogy to a power grid. When you plug an appliance or other object requiring electrical power into a receptacle, you expect that there is power of the correct voltage to be available, but the actual source of that power is not known. Your local utility company provides the interface into a complex network of generators and power sources and provides you with (in most cases) an acceptable quality of service for your energy demands. Rather than each house or neighborhood having to obtain and maintain their own generator of electricity, the power grid infrastructure provides a virtual generator. The generator is highly reliable and adapts to the power needs of the consumers based on their demand.

But, firstly, the Grid is different from cluster computing and single system parallel systems.

A cluster is made up of multiple interconnected independent nodes that co-operatively work together as a single unified resource. Unlike Grids, cluster resources are owned by a single organization and they are managed by a centralized resource management and scheduling system. That means all users of clusters have to go through a centralized system that manages allocation of resources to application jobs. Actually, many Grids are constructed by using clusters or traditional parallel systems as their nodes. For example, the World-Wide Grid, used in evaluating the Gridbus technologies and applications, has many nodes that are clusters, which are located in organizations such as AIST-Japan, N*Grid Korea, University of Melbourne, and NRC Canada. Another example of Grid that contains clusters as its nodes is the NSF TeraGrid in the US.

And moreover the Grid is different from Web services. Web services provide standard infrastructure for data exchange between two different distributed applications, whereas Grids provide an infrastructure for aggregation of high-end resources for solving large-scale problems in science, engineering, and commerce. The recent trend is to implement Grid solutions using Web services technologies. For example, Globus 3.0 version is being implemented using Web services technologies. Within the Gridbus Project, people have implemented Grid technologies such as Grid Bank and Grid Market Directory using Web services technologies. Then we can safely say that low-level Grid services are instances of Web services.

In the rest of this paper, we will introduce some Grid technologies or tools used in Grid services.

* This work is supported by NSFC basic research project: 60373023/F020103

3. GRID TECHNOLOGIES

3.1 The Globus Toolkit

The Globus Toolkit is an open source software toolkit used for building grids. It is being developed by the Globus Alliance and many others all over the world. A growing number of projects and companies are using the Globus Toolkit to unlock the potential of grids for their cause.

And the open source Globus Toolkit is a fundamental enabling technology for the "Grid," letting people share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The toolkit includes software services and libraries for resource monitoring, discovery, and management, plus security and file management. In addition to being a central part of science and engineering projects that total nearly a half-billion dollars internationally, the Globus Toolkit is a substrate on which leading IT companies are building significant commercial Grid products.

The toolkit includes software for security, information infrastructure, resource management, data management, communication, fault detection, and portability. It is packaged as a set of components that can be used either independently or together to develop applications. Every organization has unique modes of operation, and collaboration between multiple organizations is hindered by incompatibility of resources such as data archives, computers, and networks. The Globus Toolkit was conceived to remove obstacles that prevent seamless collaboration. Its core services, interfaces and protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when.

Now the Globus Toolkit has grown through an open-source strategy similar to the Linux operating systems, and distinct from proprietary attempts at resource-sharing software. This encourages broader, more rapid adoption and leads to greater technical innovation, as the open-source community provides continual enhancements to the product.

3.2 GridBank

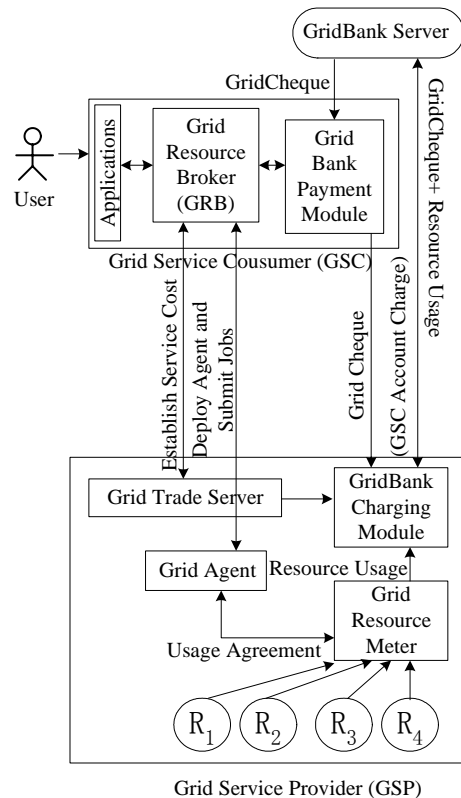
As the trend towards Internet-based distributed computing continues, large scale computationally intensive applications are executed on remote machines that are offered to provide computational services during periods when these computers are idle. Hosts connected by Internet with middleware supporting remote submission and executions of applications constitute what is called the computational Grid [1,3,4,15]. The Grid couples a variety of heterogeneous computational resources, storage systems, databases and other special-purpose computing devices and presents them as a unified integrated resource. In the global Grid environment users submit their applications to Grid Resource Broker, which discovers resources, negotiates for service costs, performs resource selection, schedules tasks to resources and monitors task executions. Resource providers advertise their services with the discovery service and run Grid Trade Service used by Grid Resource Broker to negotiate service cost. Such setup allows open market trade of computational services to take place on the Grid. Resources are offered at difference prices, and those prices are negotiated using one of several economic models from the real world [1,2].

It was observed that the utility delivered by resources is

enhanced when resource allocation is performed based on users quality-of-service (QoS) requirements/constraints (e.g., deadline and budget) [1]. In a global computing environment all users would prefer to use powerful resources, which would cause some resources to be oversubscribed and others undersubscribed. This is where computational economy and suitable service pricing strategies come into play. Resource owners are permitted to solicit an open market price in a way that achieves maximum profit and resource consumers are allowed to choose resources that meet their QoS requirements. That is, when there is less demand for resources, the price is lowered; when there is high demand, the price is raised. This helps in regulating the supply-and-demand for access to Grid resources and services.

Meanwhile GridBank can be thought of as just another source on the Grid. In other words, it is just another Grid Service Provider. Clients use the same user proxy/component to access GridBank as they use to access other resources on the Grid. A user proxy is a certificate signed by the user, which is later used to repeatedly authenticate the user to resources [7,9,15]. This preserves Grid's single sign-in policy and avoids repeatedly entering user password. Using existing payment systems for the Grid would not satisfy this policy.

And the interaction of GridBank with other Grid components can be seen from the following simple figure illustration.



In the future, GridBank system will be expanded to provide multiple servers/branches across the Grid to achieve scalability in similar manner as the currency servers in NetCash [6] and NetCheque [7] systems. It is precisely for this purpose that GridBank accounts have branch numbers. Each Virtual Organization (VO) [4,15], which is a collaboration of resources, associates a GridBank server that all participants of the organization use. If a GSC is from one VO and GSP is from another, then their respective servers will need to define

protocols for settling accounts between the branches. Moreover, if another payment system is introduced to the Grid, then that system can use different bank number and additional protocols can be defined to settle accounts between multiple banks.

3.3 Grid Market Directory

Computational Grids [1] are emerging as the next-generation computing platform and global cyber infrastructure for solving large-scale problems in science, engineering and business. They enable the sharing, exchange, discovery, selection and aggregation of geographically distributed, heterogeneous resources—such as computers, data sources, visualization devices and scientific instruments. As the Grid comprises of a wide variety of resources owned by different organizations with different goals, the resource management and quality of service provision in Grid computing environments is a challenging task. Grid economy [9] facilitates the management of supply and demand for resources. Also, it enables the sustained sharing of resources by providing an incentive for Grid Service Providers (GSPs).

It has been envisioned that Grids enable the creation of Virtual Organizations (VOs) [19] and Virtual Enterprises (VEs) [18] or computing marketplaces [20]. A group of participants with a common objective can form a VO. Organizations or businesses or individuals can participate in one or more VOs by sharing some or all of their resources. To realize this vision, Grids need to support diverse infrastructures/services [19] including an infrastructure that allows (a) the creation of one or more VO(s) registries to which participants can register themselves; (b) participants to register themselves as GSPs and publication of resources or application services that they interested in sharing; (c) GSPs to register themselves in one or more VOs and specify the kind of resources/services that they would like to share in VOs of their interest; and (d) the discovery of resources/services and their attributes (e.g., access price and constraints) by higher level Grid applications or services such as Grid resource brokers. These services are among fundamental requirements necessary for the realizations of Grid economy.

Several Grid economy models drawn from conventional markets have been proposed for organizing the Grid market [9]. They are: commodity, posted price, bargaining, tender/contract and auction models. In Grid economy models, a trusted third party, *Service Publication Directory*, is needed as a central service linking resource providers and consumers. For example, in the commodity model, resource providers publish their services to a Directory, providing service location, service type and service charge price, etc., while resource brokers query the directory and select a suitable service according to the quality-of-service (QoS) requirements (e.g. deadline and budget) of their delegating consumers.

4. CONCLUSION

Grid services provide an attractive foundation for building the services required by data center operators. They build on web services technology that is often already found in these environments, support a dynamic environment as found in active data centers, and support decentralized identity management and authentication. This latter trait is increasingly important as data centers are consolidated. What were previously geographically and organizationally distributed resources are more and more being centralized into fewer

locations and a small number of administrative domains. The grid's authentication mechanisms allow the operators to securely provide access to these centralized resources to remote users who may previously have maintained their own dedicated resources at a departmental or site-specific granularity. With the development of the Grid technologies or tools, Grid services will bring us great profits like a goldmine.

5. REFERENCES

- [1] BUYYA, R (2002): Economic-based Distributed Resource Management and Scheduling for Grid Computing, PhD Thesis, Monash University, Melbourne, Australia, April 12, 2002. Online at <http://www.buyya.com/thesis/> (23/08/2002)
- [2] BUYYA, R, ABRAMSON, D, GIDDY, J (2001): A Case for Economy Grid Architecture for Service Oriented Grid Computing. Proceedings of IPPS/SPDP '01 Heterogeneous Computing Workshop, pp. 4-18.
- [3] FOSTER, I, KESSELMAN, C (editors) (2000): The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, USA. ISBN 1-55860-475-8.
- [4] FOSTER, I, KESSELMAN, C, NICK, J, TUECKE, S (2002): The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. <http://www.globus.org/research/papers.html> (23/08/2002)
- [5] FOSTER, I, KESSELMAN, C, TUECKE, S (2002) The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal of Supercomputer Applications, 15(3).
- [6] MEDVINSKY, G, NEUMAN, B (1999): NetCash: A design for practical electronic currency on the Internet. In Proceedings of the 1st ACM Conference on Computer and Communication Security, November 1999
- [7] MEDVINSKY, G, NEUMAN, Requirements for Network Payment: The NetCheque Perspective. In Proceedings of IEEE COMPCON'95. March 1999
- [8] I. Foster and C. Kesselman (editors), The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, USA, 2000.
- [9] R. Buyya, D. Abramson, and J. Giddy, A Case for Economy Grid Architecture for Service-Oriented Grid Computing, Proceedings of the International Parallel and Distributed Processing Symposium: 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), April 23, 2001, San Francisco, CA, USA.
- [10] L. Camarinha-Matos and H. Afsarmanesh (editors), Infrastructure for Virtual Enterprises: Networking Industrial Enterprises, Kluwer Academic Press, 2000.
- [11] I. Foster, C. Kesselman, and S. Tuecke, The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International J. Supercomputer Applications, 15(3), 2002.
- [12] R. Buyya and S. Vazhkudai, Compute Power Market: Towards a Market-Oriented Grid, The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), Brisbane, Australia, May 15-18, 2001.

A Software Bus Based on the Grid Computing

Cheng YuanBin

School of Mathematics & Computer Science, Jiangnan University

Wuhan 430056 China

Email: chybin@163.com Tel: 0086-027-51257649

ABSTRACT

Grid computing removes the fixed connections between applications, servers, databases, machines, storage – every component of the grid. By treating everything in the grid as a virtualized service, grid computing can optimize resource utilization and responsiveness. However, the great advantage of grid computing can also be used to offering high quality services to the great number of users in the Internet, that is SBG—A Software Bus based on the Grid Computing. This paper states the basic characters of SBG, describes its basic constitutes, its work modes, and its basic data structures.

Key words: grid, Internet, software service, SBG, MSC

1. INTRODUCTION

Islands of computing within organizations make inefficient use of resources. Systems are slow to change and expensive to maintain. Today, grid computing addresses these problems by providing an adaptive software infrastructure that makes efficient use of low-cost servers and modular storage, Grid computing can balance workloads more effectively and provides capacity on demand. Grid computing is based on five fundamental attributes: virtualization, dynamic provisioning, resource pooling, self-adaptive systems, and unified management so that it removes the fixed connections between applications, servers, databases, machines, storage – every component of the grid. By treating everything in the grid as a virtualized service, grid computing can optimize resource utilization and responsiveness.

The original drive of grid computing comes from that how to congregate the great number of margin computing resource which dispersal in the Internet to perform great capacity computing tasks by distributed computing. But the great advantage of grid computing can also be used to offering high quality services to the great number of users in the Internet. That is the basic idea of the Software Bus based on the Grid Computing, SBG for short.

2. DESCRIPTION OF SBG AND ITS CHARACTERISTICS

The SBG makes the software resource which distributed among a range of Internet into a grid, for example, in a large enterprise, and provides a united interface to users for sharing network software services which look as the various of resource on a computer's bus and manage or use these software as same convenience as managing or using the resource on a computer's bus. So it is referred to Software Bus.

SBG usually makes up of a MSC i.e. managing and scheduling center and some S_nodes i.e. servers which providing sharing

network software services. show as fig.1.

SBG is provided with the features below:

- **Expansibility.** In theory, the number of the software nodes on the SBG can be infinitude and it is very easy to add a new node to the SBG.
- **Transparency.** On the SBG, that a user sees is only the software and their location need not to know by users.
- **QoS.** Because SBG is based on the grid, so the nice QoS provided by grid also be exhibited on the SBG.

The same points between SBG and Web service

- SBG should use the interface as same as Web browser. It because the Web browser is so welcome to people.
- SBG should use the superlinks as same as Web service.

The different points between SBG and Web service

- The essence of links is different. The links on Web is direct to a target Web page or a target Web station, it is a direct two layer link structure. While SBG is a indirect three layer link structure.
- The logical structure of the two service networks is not same. Web service is static in essence, the relation among servers and clients is one to many. While SBG service is dynamic, the relation among servers and clients is many to many.

3. SBG STRUCTURE

The SBG composed by a MSC and some S_node . The MSC is composed by a clustering server, it realizes the functions below:

3.1 Managing S_nodes

The manage to S_nodes is the foundation to ensure the SBG runs normally. The core task of the managing S_nodes is to maintenance a database for managing S_nodes . The database includes a table for software register, a table for S_node , and a list table for software - S_nodes relation. The table for software register corresponds the softwares which users see on the clients, which includes a software identifier and the number of the software as its main contents. The table for S_node includes S_node identifier and its configuring parameters and its address as its main contents. The list table for software - S_nodes relation records the distribution of all software on the SBG.

3.2 Managing client requests

The function of managing client requests is for client to get the software service requested. There are three schemes:

A) MSC act as an agent server

As Web browser has been into people's heart deeply, it is the best thing for users to use the Web browser as a client. This scheme can satisfy the demand, his scheme is also the only scheme which can realize the SBG characters while need not

change or expand WWW protocols.

But there is a fatal weakness that MSC will be a bottleneck badly in all SBG and it would bring the SBG's capability goes worsen.

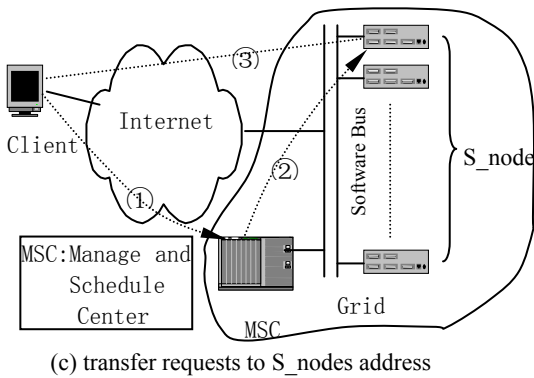
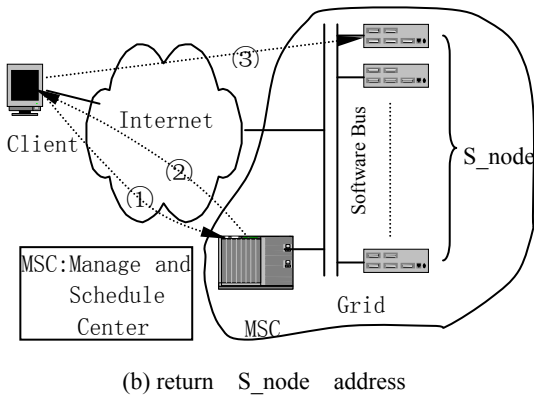
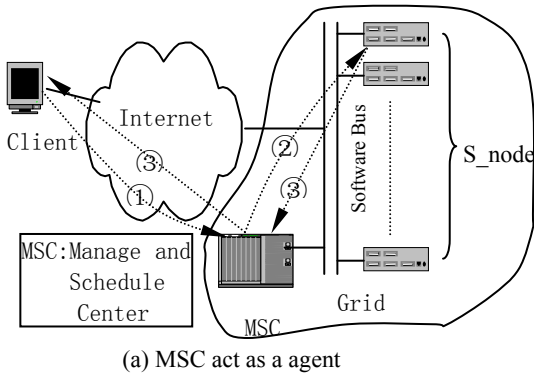


Fig.1 SBG structure

The basic procedure shows as fig.1(a), which include three steps:

- ① A client sends a request for a software service to the MSC;
- ② The MSC deals with the request and send the same request to a appropriate S_node accord the schedule strategy.
- ③ The S_node sends the service responds to the MSC and MSC transfer it to the client.

B) MSC return a S_node address to the client

The basic procedure of this scheme shows as fig.1(b), which also include three steps:

- ① A client sends a request for a software service to the MSC;
- ② The MSC deals with the request and return a appropriate S_node address to the client accord the schedule strategy.
- ③ The client connect the S_node and get the service. In this scheme, the client protocol for browser must be changed or expanded so that the client program can connect the S_node automatically according the S_address get from the MSC.

C) MSC transfer the client's request to a S_node

The basic procedure of this scheme shows as fig.1(c), which also include three steps:

- ① A client sends a request for a software service to the MSC;
- ② The MSC deals with the request and forward the request to a appropriate S_node accord the schedule strategy.
- ③ The S_node connect the client and start the service. In this scheme, the client protocol for browser and the server protocol for S_node must be changed or expanded so that the connect between the S_node and the client can be create up.

3.3 S_node schedule and load balance

Besides linking user's request to an appropriate S_node, SBG's another important function is to balance the loads among the S_nodes, in actually, load balance is a basic ruler on the S_node schedule.

The load balance problem in SBG is not same in the clustering server. In the clustering server, the services provided by every server is same, but it not always same in SBG. Therefore, we should improve the load balance arithmetic for clustering server, so that we can use the arithmetic in SBG.

In order to realize the idea above, we take some measure as below:

First, we define load points and capacity points for measuring a S_node load or capacity. The points reflects a S_node status on capacity, In another side, it also reflects the demand for a application to engross resource. So, we can do things better after we have the measuring standard.

Next, we define a memory data structure for S_node schedule and load balance, and show it in fig.2.

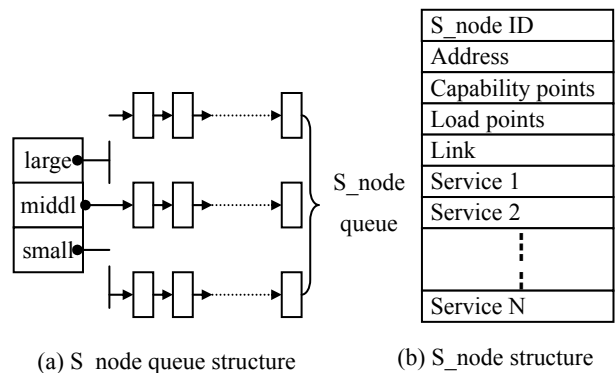


Fig. 2 schedule and load balance structure

MSC take a two level PRI balance schedule arithmetic. At the first level, we plot out S_nodes into some queues according their capacity points. For example, we plot out S_nodes into three queues in fig.2. And the frequency for every queue to be

scheduled is rest with their capacity points. At the second level, S_nodes are queued according their capacity points and FIFO ruler. When MSC receive a request from a client, it first checks if the service exists in the SBG, if so, it checks if the service exists in the S_node which is the first node in the queue which is select according the PRI balance schedule arithmetic. If the service exists in the node structure, MSC will schedule the S_node; otherwise, it checks the next node.

4. CONCLUSION

Grid computing provides us a full new mode which is powerful, it not only servers for the large scale computing task but also should server for the common server to the people. So that the Software Bus based on the Grid Computing should apply to many internet server widely.



Cheng Yuanbin is a associate Professor and a subdecanal of School of Mathematics & Computer Science, Jiangnan University. He graduated from Huazhong University of Science and Technology in 1982 with specialty of hydraulic power plant automation.

Developing Grid computing applications based on OGSA

Jing Tong, HaoChun Liu

School of Information Engineering, Wuhan University of Technology

Wuhan, Hubei, 430070, China

Email: tongjing111@hotmail.com Tel: 13545133198

ABSTRACT

The basic structure of the Open Grid Service Architecture (OGSA) and the technical specifications of the Open Grid Service Infrastructure (OGSI) were introduced in this paper. Based on the concept that service is the central of OGSA, the definition of Grid service and the general development of the Grid computing applications were discussed. Through the case study of the development of Grid computing applications using GT3, the basic theory and procedures for developing OGSA Grid computing applications were illustrated from both theory and practice.

Keywords: Grid, OGSA, OGSI, GT3, Grid service

1. INTRODUCTION

Grid is a type of integrated resources and services environment [1], and Grid computing solves problems based on Grid. With the Grid computing technology, computers all over the world can be linked into a big virtual computer system using Internet. The system has a full-scale share of computing, storage, communication, and expert resources. As a result, we have access to the abundant resources like using electric power resources. With the development of the Grid computing technology, Grid architecture has attracted more and more attention. The Grid architecture is the skeleton and soul of the Grid, and the core technology of the Grid. It describes how to building the Grid, including the definition and description of the basic components of the Grid and their function, the interrelationship of different parts of the Grid, the regulation of the integration methods, and the depiction of the efficient running-mechanism of the Grid. The Open Grid Services Architecture (OGSA) [2][3] is a newly developed Grid architecture and is also called the next generation Grid architecture. The OGSA was proposed based on the original five-layer-hourglass principle in combination with the popular

Web service technology. The OGSA has two key technologies: the Globus Toolkit 3 (GT 3) [4] and the Web service. The core of OGSA is service. Grid service is defined as a Web service that provides a set of well-defined interfaces that follow specific rules to address discovery, dynamic service creation, lifetime management, notification, and manageability. GT3 is a practical implementation of OGSA and has become the premier choice for building Grid applications. The programming design of GT3 follows the general distributed programming model, including server-side and client-side programming. In this paper, a simple case service named EchoString will be introduced to illustrate the procedure of developing Grid computing applications using GT3.

2. THE OGSA ARCHITECTURE AND OGSI SPECIFICATION

OGSA is a distributed interaction and computing architecture based around the Grid service to assure interoperability on heterogeneous systems so that different types of systems can communicate and share information.

2.1 What are the objectives of OGSA?

The objectives of OGSA are to:

- (1) Manage resources across distributed heterogeneous platforms.
- (2) Deliver seamless quality of service (QoS). The topology of Grids is often complex. Interaction of Grid resources is usually dynamic. It's important that the Grid provide robust, behind-the-scenes services such as authorization, access control, and delegation.
- (3) Provide a common base for autonomic management solutions. A Grid can contain many resources, with numerous combinations of configurations, interactions, and changing state and failure modes. Some form of intelligent self-regulation and autonomic management of these resources is necessary.

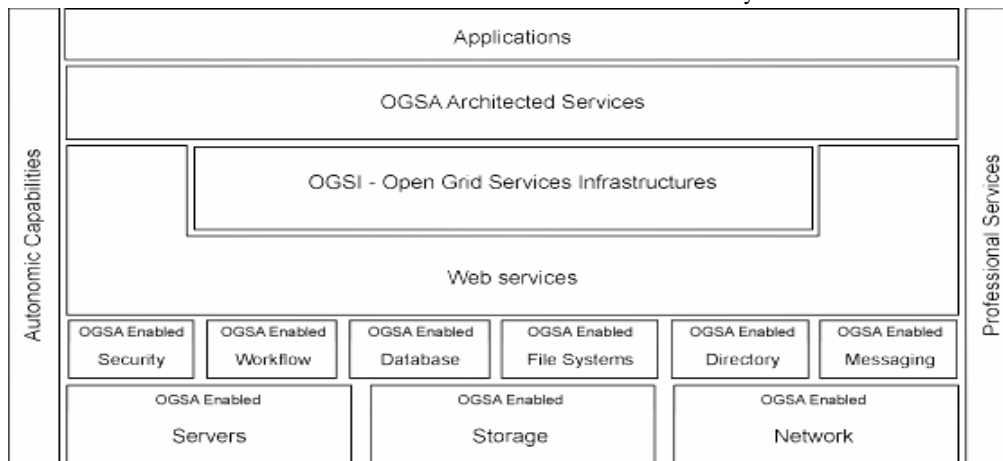


Figure 1 OGSA main architecture

(4) Define open, published interfaces. OGSA is an open standard managed by the Global Grid Forum (GGF) standards body. For interoperability of diverse resources, Grids must be built on standard interfaces and protocols.

(5) Exploit industry standard integration technologies. The foundation of OGSA is Web services.

2.2 The OGSA architecture

Four main layers comprise the OGSA architecture [5]: See Figure 1. Starting from the bottom, they are: Physical resources and logical resources layer; Web services, plus the OGSi extensions that define Grid services layer; GSA architected services layer; Grid applications layer;

Let's look at these layers, one at a time.

(1) Physical and logical resources layer.

The concept of resources is central to OGSA and to Grid computing in general. Resources comprise the capabilities of the Grid, and are not limited to processors. Physical resources include servers, storage, and network. Above the physical resources are logical resources. They provide additional function by virtualizing and aggregating the resources in the physical layer. General-purpose middleware such as file systems, database managers, directories, and workflow managers provide these abstract services on top of the physical Grid.

(2) Web services layer.

The second layer in the OGSA architecture is Web services. Here's an important tenet of OGSA: All Grid resources -- both logical and physical -- are modeled as services. The Open Grid Services Infrastructure

on top of standard Web services technology. OGSi exploits the mechanisms of Web services like XML and WSDL to specify standard interfaces, behaviors, and interaction for all Grid resources. OGSi extends the definition of Web services to provide capabilities for dynamic, stateful, and manageable Web services that are required to model the resources of the Grid.

(3) OGSA architected Grid services layer

The Web services layer, with its OGSi extensions, provides a base infrastructure for the next layer -- architected Grid services. The GGF is currently working to define many of these architected Grid services in areas like program execution, data services, and core services. Some are already defined, and some implementations have already appeared. As implementations of these newly architected services begin to appear, OGSA will become a more useful service-oriented architecture (SOA).

(4) Grid applications layer

Over time, as a rich set of Grid-architected services continues to be developed, new Grid applications that use one or more Grid architected services will appear. These applications comprise the fourth main layer of the OGSA architecture.

2.3 The OGSi specification

OGSA is composed of the two main logical components: -- the Web services-plus-OGSi layer, and the OGSA architected services layer. OGSi provides Grid services to the service layers constructed by OGSA through the extension of Web services. OGSi extends Web services by introducing interfaces and rules in the following two main areas.

(1) The dynamic and potentially transient nature of services in a Grid. In a Grid, particular service instances may come and go as work is dispatched, as resources are configured and provisioned, and as system state changes. Therefore, Grid

(OGSi) specification defines Grid services and builds services need interfaces to manage their creation, destruction, and life cycle management.

(2) There's state. Grid services can have attributes and data associated with them. This is similar in concept to the traditional structure of objects in object-oriented programming. Objects have behavior and data. Likewise, Web services needed to be extended to support state data associated with Grid services.

OGSi introduces an interaction model for Grid services. OGSi provides a uniform way for software developers to model and interact with Grid services by providing interfaces for discovery, life cycle, state management, creation and destruction, event notification, and reference management.

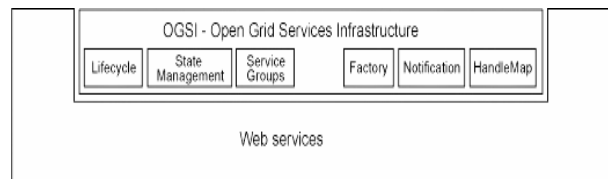


Figure 2 OGSi components

These are depicted in Figure 2. Whether a software developer is developing a Grid service or an application, the OGSi programming model provides a consistent way for Grid software to interact.

3. DEVELOPMENT OF GRID COMPUTING APPLICATIONS

3.1 Grid service

Grid service is a type of Web service, which provides a set of well-defined interfaces that follow specific conventions to address discovery, dynamic service creation, lifetime management, notification, and manageability. In OGSA, everything is regarded as Grid services. In brief, Grid service is equal to Interface/behavior plus service data. The Grid service and related technology are described in Figure 3.

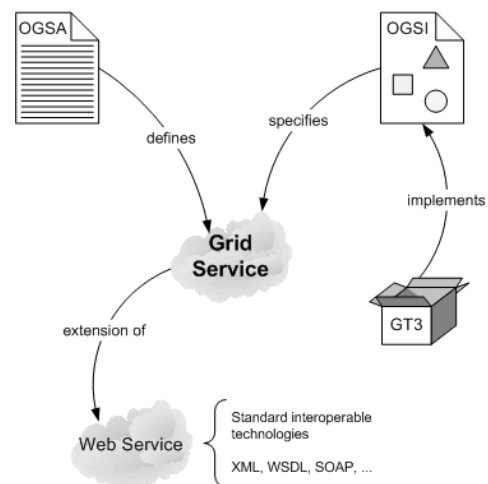


Figure 3 Grid service and its relational technologies

(1) Grid services are based on Web services. Grid services extend the interface of Web services in the dynamic, transient nature of services and the state of services.

(2) Grid services are defined by OGSA. OGSA aims to define a new common and standard architecture for Grid-based applications. OGSA defines what Grid services are, what they should be capable of, what types of technologies they should be based on, but doesn't give a technical and detailed specification.

(3) Grid services are specified by OGSi. The Open Grid Services Infrastructure is a formal and technical specification of the concepts described in OGSA, including Grid services. OGSi emphasizes more on the specific technical solutions than OGSA.

(4) The Globus Toolkit 3 is an implementation of OGSi. GT3 enables us to program applications based on Grid computing.

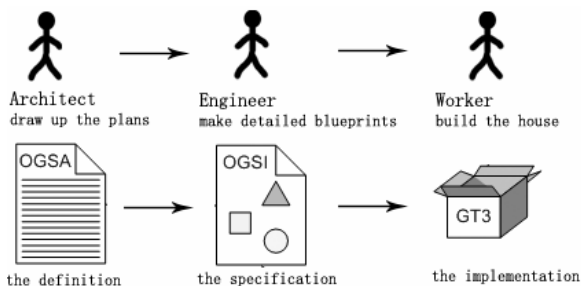


Figure 4 Contrast between building house and developing Grid applications

To assist in understanding these concepts, a case of building house are used here to illuminate the development of Grid computing applications based on OGSA. As shown in Figure 4, OGSA works as the architect to define Grid services; OGSi is the engineer, specifying the detailed blueprints; GT3 is the worker, implementing Grid computing applications.

3.2 GT3 software architecture model

3.2.1 Server-side framework

Figure 5 illustrates the components on the server side.

As shown in Figure 5, the major architecture components of the server side frameworks include the following:

(1) The Web services engine. This engine is provided by Apache AXIS framework software and is used to deal with

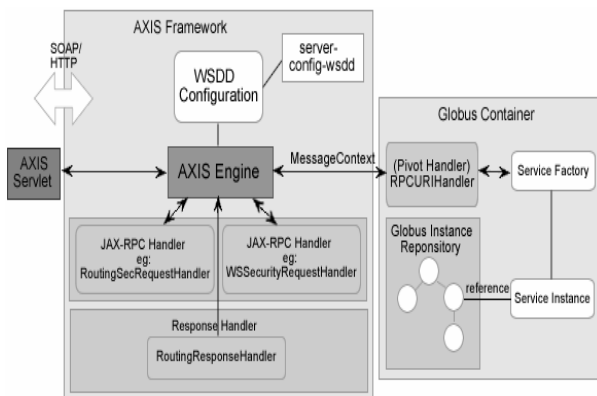


Figure 5 GT3 software framework: server-side architecture components

normal Web services behaviors, SOAP message processing, JAX-RPC handlers processing, and Web services configuration.

(2) Globus container framework. GT3 provides a container to manage the stateful Web service through a unique instance handle, instance repository, and life cycle management including service activation/passivation and soft-state management. Currently GT3 uses Apache AXIS as its Web services engine, which runs in a J2EE Web container and provides a SOAP message listener (AXIS servlet). It is responsible for SOAP request/response serialization and deserialization, JAX-RPC handler invocation, and Grid service configuration. As shown in Figure 5, GT3 container provides a pivot handler to the AXIS framework to pass the request messages to the Globus container.

This container architecture is used to manage the stateful nature of Web services and their life cycles. Once the service factory creates a Grid service instance, the framework creates a unique Grid service handle (GSH) for that instance, and that instance is registered with the container repository. This repository holds all of the stateful service instances and is contacted by the other framework components and handlers to perform services: Identify services and invoke methods; Get/set service properties; Activate/passivate service; Resolve Grid service handles to reference and persist the service.

3.2.2 Client-side framework

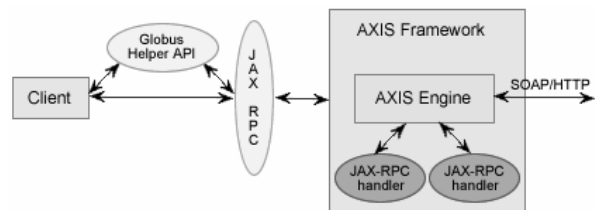


Figure 6 GT3 software framework: client-side architecture components

Figure 6 illustrates the components of the client side. As shown in Figure 6, Globus uses the normal JAX-RPC client-side programming model and AXIS client-side framework Grid service clients. In addition to the normal JAX-RPC programming model, Globus provides a number of helper classes at the client side to hide the details of the OGSi client-side programming model.

3.3 Development of a simple EchoString service using GT3

EchoString, a simple Grid service case is used here to illustrate the development of Grid computing applications based on OGSA. The function of the EchoString is simple. When client-side invokes the services of server-side, a string is appointed. After receiving the string, server-side returns the identical string to client-side.

3.3.1 Programming model of EchoString service

OGSA is a service-oriented architecture. Therefore, the function of EchoString is also provided by services in server-side. EchoStringImpl function is used to implement this service. In the architecture of OGSA, the instance of temporary service doesn't exist initially, which needs to be created by long service Factory. In this case, EchoStringImpl is a temporary service, which is created by the long service of

EchoStringFactoryImpl. Two parts need to be developed in server-side – the logic implement of temporary service EchoStringImpl and the implement of long service EchoStringFactoryImpl. OGSA is based on Web Services Description Language (WSDL) document, therefore, the corresponding WSDL services description files should be provided during the development of services. The file is EchoStringService.wsdl in this case.

The server-side programming model is shown in Figure 7. In client-side, the first thing is to get the

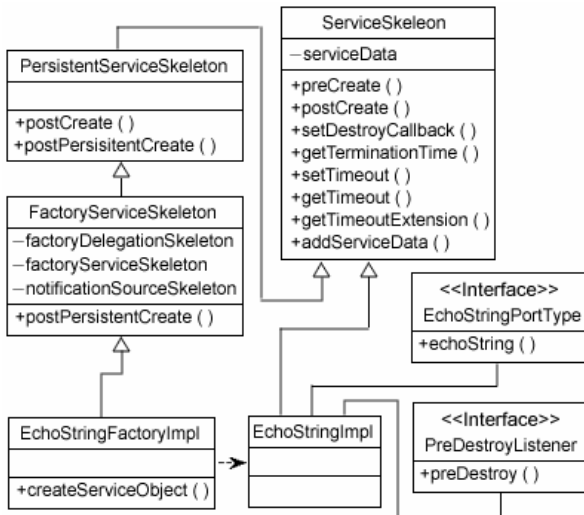


Figure 7 Server-side programming model

services description file: EchoStringService.wsdl. Consequently, the stub will be invoked according to the creation of this file. The stub in this case is EchoStringServiceSoapBindingStub.

In the end, the client-side programs invoke the EchoString service in server-side by using EchoStringServiceSoapBindingStub function.

3.3.2 Developing procedures of EchoString service

Design of server-side:

(1) Creating an interface file named EchoString.java. It is used to describe the interface of EchoString service. In this case, there is only one function called echostring, which is prepared for the generation of a WSDL file.

(2) Generating the WSDL file. A WSDL file is generated from the Java interface by using an Apache Axis tool called Java2WSDL.

(3) Generating the Stubs. After obtaining the WSDL file, the next step is to generate Stubs, which are used to serialize and deserialize data from the XML messages.

(4) Implementing the service. The first part is the logical implementation of EchoString service. In this part, the string sent from the client-side is sent back using the sentence of return in() in the echostring function. A case of EchoString service is generated thereafter. When there is request from the client-side programs for the implementation of Echostring service by the server container, the server container invokes the createServiceObject function from the EchoStringFactoryImpl class. This function will generate a case of EchoString that will return to server container using

return newEchostringImpl().

(5) Deployment of the Grid service. Up to this point, we have written all the java code our Grid Service needs. We have a service interface, a WSDL file, a bunch of stub files, and an implementation of the service. The task of this step is to put all these pieces together, compile, and deploy them into a hosting environment. Three steps are involved, creating a service deployment descriptor file, creating a Jar file, and deploying them into hosting environment.

Development and deployment of server-side is finished up to here.

Design of client-side:

In client-side, the first step is to obtain the EchoStringService.wsdl, the WSDL service description file of EchoString. The stubs will be generated in the same way as that of the step (3) for server-side thereafter. A client-side program Echo.java will be created in the end.

3.3.3 Brief summary of developing Grid computing applications using GT3

Through the introduction of OGSA and the development of Grid computing applications using GT3, we can conclude as follows:

(1) GT3 programming follows the common distributed computing model, i.e. proxy-stub model. The difference is that OGSA allows the independent implementation of Factory by service provider. As a result, service providers can easily and independently manage the instances cases of service.

(2) OGSA is document-oriented. The only link between server-side services and client-side programs is WSDL services description file. Service providers have to provide the corresponding service WSDL file during the development of services. In this file, the interface, invocation method of services and the binding information between service invocation and lower-level communication protocol are described in details. After receiving the WSDL file, the client-side generates the stubs for service invocation. The stubs are used to complete the invocation of services. In addition, based on the WSDL file, it's possible to separate the top-layer services from the bottom-layer communication protocol. The service providers can provide many methods to bind service invocation and communication protocol, from which the client-side can choose the proper one to complete the invocation of service. It is even possible to delay the invocation process by choosing the right binding method.

(3) Besides ensuring the accurate and efficient invocation of service, the service container also has to fulfill the management function of the integral characteristics of the Grid such as service registering, service quality controlling, and message, affair and security management. Under the insurance of these functions, the service providers can concentrate on the logic implement of services by leaving the control of the service running environment to the Grid middleware. Therefore the amount of work of the service provider could be reduced.

4. CONCLUSIONS

The Grid computing is a rapidly developing field, and the Grid architecture is the core technology of the Grid constructing.

OGSA, one of the most important new architectures, is introduced in this paper. Taking the advantage of existing architecture and technology, OGSA proposed a service-oriented architecture by using the uniform frame of Web service. The major characteristic of this architecture is that it extends the Grid computing from the traditional science and engineering computing based research fields to social and economic activities such as enterprise and cooperation. This greatly extended the application of the Grid computing. To get an idea about the general development of Grid computing, the concept of Grid service and its implementation technologies are also discussed in details from the aspect of developing Grid computing applications based on OGSA. When implementing OGSA, GT3 has become the premier choice for building Grid applications. The programming design of GT3 follows the general distributed programming model, including server-side and client-side programming. Through the case study on developing a Grid computing application using GT3, the implementation procedure of GT3 programming is illustrated in this paper to facilitate the understanding of the theory of implementing Grid computing applications based on OGSA.

5. REFERENCES

- [1] I.Foster , C.Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, San Fransisico, CA, 1999. <http://www.gridforum.org/>
- [2] Ian Foster, Carl Kesselman, Jeffrey M. Nick , Steven Tuecke, The Physiology of the Grid -- An Open Grid Services Architecture for Distributed Systems Integration, <http://www.globus.org/>
- [3] Steven Tuecke, Karl Czajkowski, Ian Foster, Jeffrey Frey, Steve Graham, Carl Kesselman, Grid Service Specification, <http://www.gridforum.org/>
- [4] www.globus.org, The Globus Toolkit 3 Programmer's Tutorial, <http://www.casa-sotomayor.net/gt3-tutorial/>
- [5] Jay Unger, Matt Haynos, A visual tour of Open Grid Services Architecture, <http://www-900.ibm.com/>

Research on the Architecture-based Adaptive Grid Application

Guoyou Zhang¹, Yinzhang Guo¹

¹Computer Department, Taiyuan Heavy Machinery Institute
Taiyuan, Shanxi, 030024, China

Email: guoyouzh@i618.com.cn Tel.: +86 (0)351-6220101

ABSTRACT

Grid application has emerged for the distributed computing infrastructure. But few consult it from software architecture. In this paper, the analysis of grid infrastructure and its application was prompted from the view of software architecture firstly. The architecture of grid is based on three-layer. Then according the nature of grid, we denote the architectural model of adaptive grid application. The mechanism of adaptation management is by divided it from application and can be generalized from particular system.

Keywords: grid computing, software architecture, adaptive grid computing, adaptive software.

1. INTRODUCTION

“Grid” was coined in the middle 1990s to demote a proposed distributed computing infrastructure for advanced science and engineering ^[1]. A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and expensive access to high-end computational capabilities. A computational grid is concerned with large scale pooling of resources. Such pooling requires significant hardware infrastructure to achieve the necessary interconnections and software infrastructure to monitor and control the resulting ensemble.

The concept of sharing distributed resources is not new. However, a combination of technology trends and research advances makes it feasible to realize the vision, commonly to as the grid. The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations ^[2]. Sharing means highly controlled, with resource providers and consumers defining just what is shared, including who is allowed to share, and the conditions under which sharing occurs. VO (virtual organization) is formed by a set of individuals and/or institutions defined by such sharing rules. Current distributed computing technologies do not address the concerns and requirements. For instance, current Internet technologies address communication and information exchange among computers but do not provide integrated approaches to the coordinated use of resources at multiple sites for computation. Business exchanges focus on information sharing. Enterprise distributed computing technologies such as CORBA and Enterprise Java enable resource sharing within a single organization, which cannot satisfy the demand of VO.

Considerable progress has been made on the construction of Grid application infrastructure ^[3,4,5,6]. But few consider Grid Construction and its application from the view of software architecture. In this paper, we present the architecture of Grid application firstly. In section 2, we show the architecture of the adaptive grid. At last we depict our architectural model in

practice.

2. ARCHITECTURE OF THE GRID

The establishment, management, and exploitation of dynamic, cross-organizational VO sharing relationships require new technology ^[2]. One of expressive way is building it from the architecture view. A standards-based open architecture facilitates extensibility, interoperability, portability, and code sharing. Among the nature of Grid architecture, interoperability is the key issue to be addressed. Interoperability can guarantee that sharing relationship can be initiated among arbitrary parties, accommodating new participants dynamically, across different platforms and programming environments.

Our view of architectural Model is three layer-based, which is listed below.

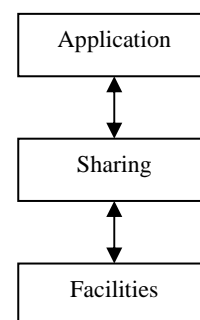


Figure 1 Architectural Model of Grid

The first layer is facilities. Many existing protocol and implementation has existed for the network communication, control and collaboration, for example, CORBA, Open Group's Distributed Computing environment. But they are burdensome and inflexible if they are used to build grid application. Hence such protocol and implementation should be selected partly or modified to meet the demand of VO. much additional facilities May be compound to this layer despite of existing protocol and implementation.

The middle layer is sharing. One of most benefit and intention is sharing resource sharing resource include the aspect of communication and control access of resource. It performance two task: for the grid communication securely and expediently and coordinate the multiple resource, so it can be divided into two sublayers, one for connectivity and the other for Coordinating Multiple resource. Connectivity sublayer provides core communication and authentication protocol. Authentication protocol required for grid-specific network transactions. Communication requirements include transport, routing and naming. Some protocols such as TCP/IP already exist. It can be taken benefit of them, but not for all. Authentication solutions for VO environment should have the

characteristics of Single sign on Delegation, Integration with various local security solutions and user-based trust relationships^[7].

The top layer is application. It includes sophisticated frame works and libraries .For example Common Component Architecture, CORBA, Workflow system etc. it is consisted of all kind of application in various field and domain.

This three layer-based grid architecture is the overview of grid from software architecture.

A standard-based open architecture of the grid facilitates extensibility, interoperability, portability and code sharing^[2]. Among the nature of architecture, interoperability is the key issue to be addressed, which guarantees the need of sharing relationship can be initiated among arbitrary parties, accommodating new participants, dynamically across different platforms and programming. Facilities layer should provide the resource to which mediated by grid protocols here

3. ARCHITECTURE OF ADAPTIVE GRID

Self-Adaptive software can dynamically decide and adjust its output and state in terms of the environment that interacts to. Laddaga^[8] defines self-adaptive software as:

“Software that evaluates and changes its own behavior when the evaluation indicates that it has not accomplishing what it is intended to do, or when better functionality or performance is

possible.”

Self-adaptive system can evaluate its behavior and decide the change of system state. For the Grid computing, it offers a wide range of distributed resources to application. So the adaptation management is an important requirement for Grid applications. For instance, Grid applications should be able to adapt themselves during runtime to handle things such as resource variability and system faults, and should be able to adapt automatically, within minimal human intervention.

Software architecture views system as the combination of components and connectors together with its configuration. It can be used to build the system from architectural model. Many architectural styles are generalized such as pipe-filter, blackboard and shared information system. For the adaptive system, the use of architectural model can have a number of benefits:

- A global perspective on the system can be made by the architecture and the system can determine some global changes, such as adding or changing components, to achieve some property.
- Self-adaptive can dynamically change its behaviour during runtime. Architectural model can clarify the constraint condition and help to ensure the validity of any change.
- Using the architectural model and separate the adaptation management of system, it can make the adaptation mechanism separate from the specific system, which makes the reuse achievable.

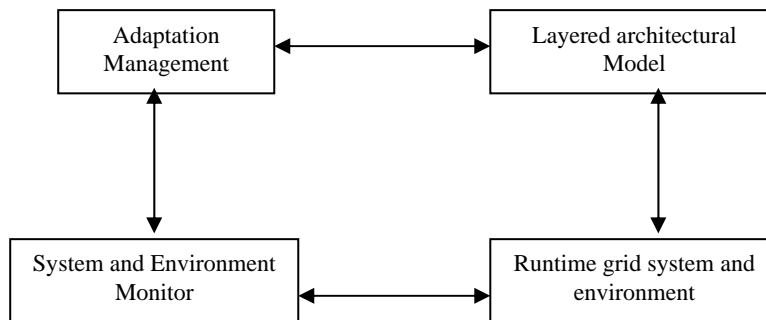


Figure 2. Architectural Model of Self-adaptive Grid System.

In this model, we divide the adaptation management from the grid application. The block of Runtime grid system and environment consists of the application itself together with its operating environment. The block of Runtime and Environment monitor collects the information that is useful to adaptation. This information is filtered and present to the adaptation management. Layered architectural model interact with the runtime grid system and verify the adaptation from the adaptation management. Adaptation management receives the information from the system and environment monitor, and makes decision for architectural Model to adjust the system runtime behaviour.

4. ARCHITECTURAL MODEL IN PRACTICE

To illustrate this architectural Model, we develop a drawing

editor Called GCWC for the adaptive experiment. Our experiments on the adaptive grid application focus mainly on the remote data store and the transportation bandwidth. This drawing editor can draw the basic shape from client system and store in the remote server or other similar system. Such system can change its server according transportation bandwidth to the server available on the net, which lead transferring the data for convenience and secure as possible. In our experiment, the secure of grid application is considered. User logging on a Server can pass through the whole group and share the grid resource.

To achieve this target, some extra operations are introduced. For the system and environment monitor, the measurement of bandwidth and filter function is needed. For the layered architectural model, the analysis of application constraint and adjustment of system is necessary. For the adaptive management, the instructions of guiding layered architectural

model should be prompted.

5. REFERENCES

- [1] Foster, I. and Kesselman, C. (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.
- [2] Foster I. and Kesselman C. The Anatomy of the Grid: Enabling Scalable Virtual Organizations Supercomputer Applications, 2001.
- [3] Benger W., Foster I., Novotny J., Seidel E., Shalf J., Smith W. and Walker P. Numerical Relativity in a Distributed Environment. In Proc. 9th SIAM Conference on Parallel Processing for Scientific Computing, 1999.
- [4] Brunett S., Czajkowski K., Fitzgerald S., Foster I., Johnson A., Kesselman C., Leigh J. and Tuecke S., Application Experiences with the Globus Toolkit. In *Proc. 7th IEEE Symp. On High Performance Distributed Computing*, 1998, IEEE Press, 81-89.
- [5] Johnston W.E., Gannon D. and Nitzberg B., Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid. In Proc. 8th IEEE Symposium on High Performance Distributed Computing, 1999, IEEE Press.
- [6] Stevens R., Woodward P., DeFanti T. and Catlett C. From the I-WAY to the National Technology Grid. *Communications of the ACM*, 40(11): 50-61. 1997.
- [7] Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J. and Welch, V. Design and Deployment of a National-Scale Authentication Infrastructure. *IEEE Computer*, 33(12):60-66. 2000.
- [8] Laddaga R. Active Software. in First International Workshop on Self-adaptive software (IWSAS2000) 2000.



Guoyou Zhang is an engineer and a member of network division in computer department, Taiyuan Heavy Machinery Institute. He graduated from Jiaozuo Mining Institute for bachelor in 1994, and from TaiYuan Technology University for master in 2003. His research interests are software architecture and grid computing.

Grid Computing Resource Management Scheduler Based on Genetic Algorithm *

Hao Tian, ZhouZude, Liu Quan

School of Information Engineering, Wuhan University of Technology

Wuhan, Hubei, 430070, China

Email: haotian@mail.whut.edu.cn Tel.: 13317182977

ABSTRACT

Computation grid is one of the research hotspots in high performance computing area. The grid resource management is more complex than in other common systems for grid system consisted by mass heterogeneous resources, which are complicated, dynamic and autonomous. Scheduling policy is the difficulty of it. Firstly, the authors analyze the characteristics of grid task, and introduce the model of it. Secondly, the authors bring forward a hierarchy grid resource management scheduler, expound the ideas in designing the model and give the detailed description of it. Finally, the authors adopt GA in the scheduling policy and particularize the principle and the function of this algorithm as well as giving the concrete plan to put every step of the scheduling policy into practice. This model can provide a uniform upper-level management framework for grid resource management and it also can realize the global optimum scheduling. It is an effective approach for grid scheduling.

Keywords: Grid, Genetic Algorithm, Model, Policy, Resource Scheduling

1. INTRODUCTION

Grid technology is an important information technology, which has growing up globally during the recent years. Its object is to build a universal and mass computing process virtual system, which is consisted by several distributed resources including computation hosts, network bandwidth and data centers.^[1] Grid technology has broad prospect of application in many fields, such as commerce, transportation, meteorology and education. There are great needs for high performance grid in many departments and enterprises which referring to scientific research, development and education.

Resource management is one of the key technologies in grid computing. It couples grid resources logically as a single integrated resource for users. Users can communicate with the resource agent directly without considering the complexity of grid resources and computation grid. Generally speaking, grid resource management system has three kinds of basic services: resource distribution, resource detection and resource scheduling.^[2]

Resource scheduler is an important part of grid resource management. The efficacy and acceptability of resource management mainly depend on its resource scheduler. Resource scheduler allocates needed resources to the corresponding requests, including cooperation allocation through different nodes. However, resource scheduling is

becoming a complicated problem because of the heterogeneous and dynamic characteristics of grid system as well as the different needs for the resources of the application which is applied in grid system.

The paper brings forward a hierarchy grid resource management scheduler and introduce genetic algorithm (GA) in scheduling policy.

2. MAIN MODELS AND SCHEDULING METHODS OF GRID RESOURCE MANAGEMENT SYSTEM

Architecture of the model of resource management mainly depends on the number of the resources needed to be managed and the tasks needed to be scheduled; it also rests with whether the resources are in a single area or in several areas. There are three main models of resource management today:^[3]

(1) Centralized Management Model. The model can be used to manage resources in a single or several areas, but it can only support consolidated scheduling policy, it's suitable for group management system and alignments system. As grid resource management scheduler must obey the local strategy of every resource owner, it is not suited to the model.

(2) Distributed Management Model. In the model, it is the interaction among every resource management system to decide which resource is suitable for the task. In the precept, every system is equal to others, there is no mechanism to manage the whole system, so the model has high extensibility and fault tolerance. As resource owner can define the policy of scheduler, this kind of distributed mechanism fits for grid system. However, because long-distance tasks and resource status are useless on single position, the model can not bring the best scheduling method, and it is difficult to be realized in grid.

(3) Hierarchy Management Model. It's a mixed model (combination of centralized and distributed models), but it's more like a centralized model. It synthesizes the characters of the two models; it not only can obey the local scheduling policy of resource owner, but also can manage whole system with the best scheduling method, so it's suitable for grid system.

Grid scheduling policy is the core of grid scheduling. There are so many special grid scheduling policies such as the scheduling policy based on DAG, the scheduling policy based on SA, MET, MCT, Min-min (Max-min) and so on. The paper adopts the scheduling policy based on GA.

* Supported by: national key scientific and technological project (No: 2003BA103C)

3. GRID TASKS ANALYSIS

3.1 Characters and Model of Grid Task

Different grid resource scheduler and different scheduling policy have different view about grid task. Under the condition of taking complete time as optimization target, in the process of grid scheduling, we regard every user's request as a meta task, and partition a meta task into several independent tasks, viz. regard a meta task as a union of independent and one-off tasks can be scheduled at a time, but we don't exclude the dependence in every task, viz. subtask. We regard a task as a combination of a data transfer subtask and a computing subtask, and we only consider the two subtasks using correlated resource to execute task. The model of meta task as shown in Figure 1.

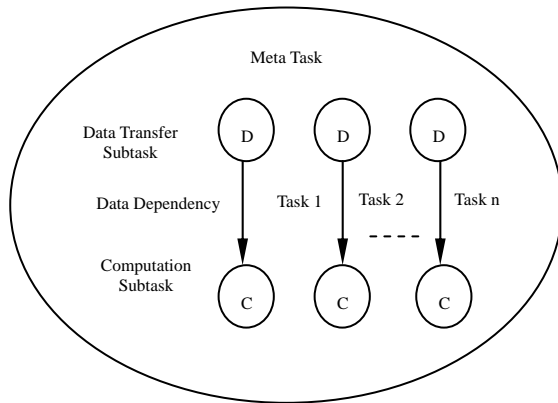


Figure 1 Model of Meta Task

In Figure 1, data transfer subtask means the communication needed in computation task, viz. the part that needs expending store time and communication time. Computation subtask is the part of expending computing time. In fact, it's reasonable to partition task in current grid system and computing module. In the process of grid scheduling, a group of computers or mainframe computers are usually regarded as a computation resource (node), because they are autonomous systems based on SSI (Single System Image), they have sound scheduling measures. After user's task is submitted to grid, grid-scheduling program locates the data the task needs and assigns the task to the most suitable node, and the node executes the task after it gets the data. Changing the scheduling policy of a computer group or a mainframe is impossible and unnecessary, [5] so we do not discuss whether there is a subtask in the two subtasks.

3.2 Grid Task and Grid Resource

From what we have analyzed above, we can know that after executing each scheduling, every task relates to a data center and a computation host, viz. relates to a group of grid resources. Data center first transfers the data that computation subtask needs to computation host, and computation host begins to execute the computation subtask as soon as the data transfer has finished, so the complete time of a task can be regarded as the sum of the complete time of its data transfer subtask and the complete time of its computation subtask.

4. MODEL OF RESOURCE MANAGEMENT SCHEDULER

4.1 General Frame and Description of the Model

Figure 2 shows the frame of our model, it includes four parts, viz. task partition module, scheduling decision module, information collection module and grid resource module.

Task partition module partitions a grid application (a meta task) into several independent tasks.

Scheduling decision module uses the scheduling policy based on GA to distribute every task to corresponding computation host and data center.

Information collection module is composed of MDS and NWS. MDS^[6] is a grid information management system, it is used to collect and issue the state information of a system, we can gain much information from it: union of available resource nodes, attribute of every node, such as type of processor, speed of processor, amount of available processors etc. NWS^[7] is a distributed monitor system, it is specially designed to monitor resources in existence and network state, it can provide short-term network capability forecast, and it works on every resource node so as to provide real time monitoring. We can get such data through NWS: availableCPU, currentCPU, connectTimeTcp, bandwidthTcp, latencyTcp etc. Information collection module collects information about grid resources, and feed it back to scheduling decision module, so as to provide evidence for scheduling policy.

Grid resource module includes several heterogeneous computation hosts, data centers and network bandwidth.

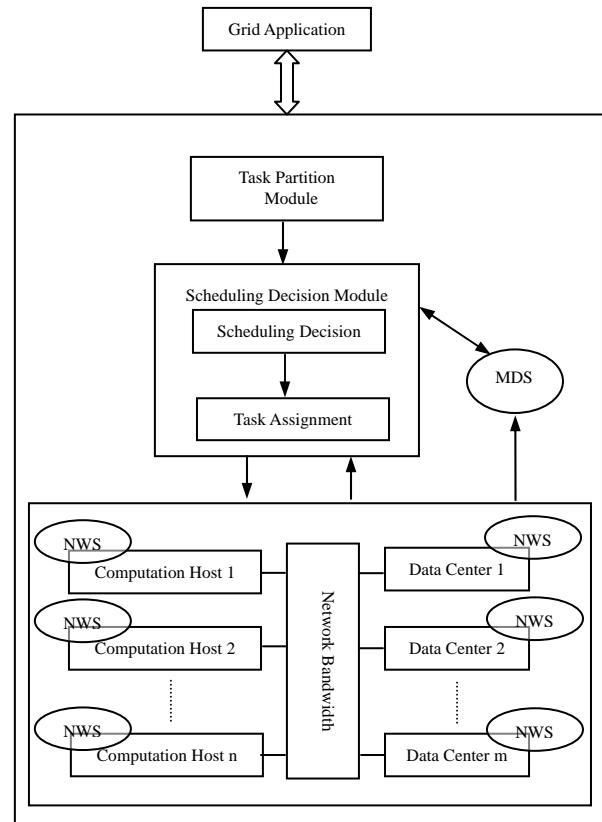


Figure 2 Model of Grid Resource Management Scheduler

4.2 Scheduling Policy of the Model

We use the scheduling policy based on GA. GA is a method of stochastic optimization and search, and it has the self-adapted search ability which has potential ability of learning [8]. It expresses solution of problem as chromosome, before executing algorithm, it produces a group of chromosomes, viz. tentative solution, and puts the tentative solution into the environment of problem, then chooses the chromosomes can fit the environment from the group on the principle of survival of the fittest, produces a new generation group of chromosomes fit the environment better through crossover and mutation. Anagenesis generation to generation, there will be a new group of chromosomes fittest the environment, viz. the best solution of problem. The principium of GA can be described as such steps:

- ① Choose an initialization population
- ② Determine the fitness of each individual
- ③ Perform selection
- ④ Repeat
- ⑤ Perform crossover
- ⑥ Perform mutation
- ⑦ Determine the fitness of each individual
- ⑧ Perform selection
- ⑨ Until some stopping criterion applies

5. SCHEDULING POLICY BASED ON GA

5.1 Description of the Problem

Based on previous discussion, we suppose a meta task is T , and it can be partitioned into l independent tasks, grid computation system is composed of m heterogenous computation hosts C ($C = \{C_0, C_1, \dots, C_{m-1}\}$) and n heterogenous data centers D ($D = \{D_0, D_1, \dots, D_{n-1}\}$), every computation host can get data it needs from any data center, then there are $m \times n$ communication lines (network bandwidths) in whole grid system, viz. $m \times n$ groups of grid resource. We express these groups as a $m \times n$ matrix R ; express the transfer time from a data center to a computation host as a $m \times n$ matrix DT ; express the computing time of a task on a computation host as a $m \times l$ matrix CT .

This shows, the essential of grid resource scheduling is distributing l independent tasks to $m \times n$ groups of grid resource in order to minimize complete time of a meta task and use grid resources sufficiently.

As the model shows, the value of matrix DT can be provided by NWS directly. The value of matrix CT can be calculated by:

$$CT(T_i, R(j, k)) = speed(C_j) \times \frac{processors(C_j)}{load(C_j) + 1} \quad (0 \leq i \leq l-1, 0 \leq j \leq m-1, 0 \leq k \leq n-1) \quad (1)$$

Where

- $speed(C_j)$ is the processor speed of node j ;
- $processors(C_j)$ is the amount of processors of node j ;
- $load(C_j)$ is the average load of node j .

The three indexes can be gotten by NWS and MDS. Complete time of a task is as the following:

$$T_i T = DT(j, k) + CT(T_i, R(j, k)) \quad (0 \leq i \leq l-1, 0 \leq j \leq m-1, 0 \leq k \leq n-1)$$

Complete time of a meta task is :

$$MTT = \max \{T_i T, i = 0, 1, 2, \dots, l-1\} \quad (3)$$

So the task of scheduling is to realize the best assignation of grid tasks on groups of grid resource to minimize MTT.

5.2 Initialization

The chromosomes coding technology we use is subsection according to the amount of tasks, a section expresses a task, it forms from a task mark T_i and a resource group mark $R(j, k)$, it means that task T_i is executed by resource group $R(j, k)$. In order to create a initialization population, we distribute tasks to resource groups equally and stochastically.

5.3 Fitness Function

Choosing a proper fitness function can evaluate every iterative solution well. Here we suppose fitness function is $f = T_i T' / T_i T_m$, $T_i T'$ is the complete time of task T_i with current scheduling policy, and $T_i T_m$ is the complete time of task T_i with the best scheduling policy.

5.4 Selections and Anagenesis

We calculate the fitness function of task T_i of every generation, keep it if it fits the demand of convergence, then choose other tasks into choice union and realign their orders in chromosome.

Anagenesis is used to generate next generation chromosome. In the chosen sections of chromosome, it first tries different combinations of tasks within changeable limits, viz. tries to distribute the tasks to random grid resource groups once again, then calculates their complete time, and chooses the best scheduling combination, consequently finishes a time of anagenesis, the new chromosome will be the origination of next anagenesis, in this way, the global optimum solution will be found.

6. CONCLUSIONS

On the basis of current research, we take the hierarchy management model as prototype, use GA in scheduling policy, bring forward a mended grid resource management scheduler based on GA. It has good expandability, can realize global optimum scheduling, it is an efficient plan of grid resource management scheduling.

7. REFERENCES

- [1] Baker M, Buyya R, Laforenza D. The Grid: International Efforts in Global Computing. In: Intl. Conf on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR 2000), 1, Aquila, Rome, Italy, July 31-August 6. 2000
- [2] Krauter K, Buyya R, Maheswaran M. A Taxonomy and Survey of Grid Resource Management Systems. Software Practice and Experience, 2002, 32(2):135—164
- [3] Rajkumar Buyya, et al. An Economy Driven Resource Management Architecture for Global Computational Power Grids[C]. The 2000 International Conference on Parallel and Distributed Processing Techniques and

Applications(PDPTA 2000), Lasvegas, USA, 2000, 26-29.

- [4] Tracy D Braun, Howard Jay Siegel et al. A Comparison of Eleven Static Heuristic for Mapping A Class of Independent Tasks onto Heterogeneous Distributed Computing Systems[J]. Journal of Parallel and Distributed Computing, 2001; 61(6):810~837
- [5] Zha Li, Xu Zhiwei, Lin Guozhang, Liu Yushu. Grid Task Scheduling Simulations Based on Simgrid. Computer Engineering & Application, 2003.14
- [6] Globus Project. <http://www.globus.org>
- [7] Network Weather Service. <http://nms.cs.utk.edu>
- [8] IEEE Transactions on Parallel and Distributed Systems, 2001.12(9)



Hao Tian, male, was born in NOV. 1977, is a postgraduate in the School of Information Engineering, Wuhan University of Technology. His research direction is system and signal processing.



Zhou ZuDe, male, professor, tutor of doctor, is the President of Wuhan University of Technology, his research interests are CNC Theory and technology, Intelligent Control, Digital Manufacturing, Reliability and Fault Diagnosis of the Modern Manufacturing Systems and etc.

The Scheme of Synthetic Forces Express Based on Vega Grid

Cheng Zhou Qingping Guo
 School of Computer Science and Technology, Wuhan University of Technology
 Wuhan, Hubei 430070, China
 Email:zc_overwhelming@sina.com Tel.: +86(0) 27 50855451

ABSTRACT

The Synthetic Forces Express is developed by USA for military simulation. It is important to our nation for modern military development. The Vega Grid of our own is really a perfect solving scheme for SF Express. The rest of the paper is organized as follows: In Section 1, we give an overview of the Synthetic Forces Express. In Section 2, we introduce the architecture of Vega Grid. The Vega Grid Based Configuration for SF Express will be discussed in Section 3, including task disassembling and syncretizing with Globus. At the conclusion of this paper, we also point out the aspects for further work.

Keywords: SF Express, Vega Grid, Globus

1. INTRODUCTION

The Synthetic Forces Express^[1] (SF Express) project began in 1996 developed at the California Institute of Technology with DARPA funding with the goal of investigating the use of high-performance computers as a means of supporting very large-scale Distributed Interactive Simulations (DIS). A specific charter for SF Express was to use supercomputers for demonstrating scalable communications architecture, supporting tens of thousands of vehicles.

In 1996, A simulation of 10,000 vehicles was achieved using 1,024 total processors on one Intel Paragon parallel supercomputer. In 1997, SF Express was refined and extended to include multiple high-performance computers. A simulation of 50,000 vehicles was achieved using 1,904 total processors on six computers at sites distributed across seven time zones.

It soon became apparent that issues other than scalable communications needed to be addressed in order to improve the functionality and validity of large-scale distributed simulation experiments. Such issues include: scenario distribution, resource configuration, resource management, information logging, monitoring, and fault tolerance. To address some of these issues, they integrated services provided by the Globus^[2] Metacomputing Toolkit. And on March 16, 1998, a record-breaking simulation was conducted using a 100,298 vehicle entity-level simulation - the largest, distributed, interactive battlefield simulation to date. This simulation was executed on 1,386 processors that were distributed over 13 computers among nine sites that spanned seven time zones. Global and local services were decoupled, allowing the application to run in a flexible, resource-aware environment. They continue to incorporate emerging computational grid tools and techniques into the distributed interactive simulation environment, bringing increased benefits of pervasive and dependable access to high-end computational capabilities.

Thus it can be seen, Grid^{[3] [4]} realization scheme is vital to

the result of SF Express. Our nation also needs scalable computing for military simulation in this way. The Vega Grid conducted at Institute of Computing Technology, Chinese Academy of Sciences is really a perfect solving scheme for it.

2. GRID COMPUTING AND THE VEGA GRID

2.1 What is Grid?

Ian Foster et al^{[5] [6] [7]} offered several definitions grid: "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities". Grid computing is concerned with "coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations (VO). ... [A VO is] a set of individuals and/or institutions defined by [some highly controlled] sharing rules."

"A grid is a system that (1) coordinates resources that are not subject to centralized control, (2) using standard, open, general-purpose protocols and interfaces, (3) to deliver nontrivial qualities of service."

2.2 The Vega Grid Project^{[8] [9] [10] [11] [12]}

The Vega Grid is a research project conducted at Institute of Computing Technology, Chinese Academy of Sciences. It aims to learning fundamental properties of grid computing, and developing key techniques that are essential for building grid systems and applications. The Vega Grid team currently consists of more than 150 people, and is conducting research work in the following areas:

·**Dawning 4000 Superservers:** Terascale grid enabling clusters on Linux/Intel and AIX/PowerPC platforms.

·**Vega Grid Software Platform:** This work includes research on grid system software, grid application development tools, and grid user interface. The objectives are to enable resource sharing, collaboration, service composition, and dynamic deployment, utilizing open standards such as OGSA, Globus and web services.

·**Vega Information Grid:** Research on enabling technology for information sharing, information management, and information services in an ASP environment or a wide-area enterprise environment.

·**Vega Knowledge Grid:** Research on knowledge sharing, knowledge management, and knowledge services in a wide area Web environment.

Central to the Vega Grid project is the *VEGA Service Grid* principle. The Service Grid concept abstracts three aspects of applications requirements: (1) The Vega Grid should enable user visible services, not just providing an infrastructure. (2) Service is the main mechanism for users to interact with grid.

(3) The criteria used to evaluate grid functionality and performance should evolve from traditional criteria (e.g., speed and throughput) to service-oriented criteria, such as Service Level Agreement (SLA). To realize the service grid concept, the Vega Grid project follows the following VEGA principles:

Versatile Services. The grid should have the ability to support various services and resources, not just scientific computation. The Vega Grid project aims to satisfying the minimal common requirements of various grid applications.

Enabling Intelligence. The grid should support intelligent computing, such as automatic production of information, knowledge and services. However, the grid itself is not the intelligence provider, but it provides enabling technology to assist developers and users to achieve intelligent grid applications.

Global Uniformity. From the user's viewpoint, the grid can be viewed as a single virtual computer, supporting single system image (e.g., single sign-on). Heterogeneous resources among geographically distributed grid nodes should form a uniform, connected, inter-operable resource pool, instead of many islands.

Autonomous Control. The grid should not be ruled by a central administration. All components can freely join or leave the grid at their own will. A resource provider has full control of its resource exported, and a user can use resources as he likes within the purview of his right.

2.3 The Vega Grid Architecture^[9]

A three-layer architecture of the Vega Grid is illustrated in Figure 1. At the grid hardware layer, we are developing Dawning 4000 superservers, which are clusters with enabling technology to support grid platforms and applications. Other components at the grid hardware layer include a client device and a router. The *Vega Grid Client* is an easy to use client device for grid users. The *Vega Grid Router* enables application-level connectivity and allows resources to be efficiently deployed and discovered. The grid software platform layer includes grid system software and middleware, such as Globus, OGSA, web services, and other commercial grid software, as well as technologies developed by the Vega team. The application layer includes various application software servers, such as database servers, web servers, and business application servers. The Vega Grid adds two new components at this layer, one at the client side and one at the server side. The Vega Grid "Browser" is different from a traditional web browser, in that it allows users to write to and to operate the grid. The Vega Grid Server (the GSML server) is a portal to the grid, which provides a logically single entry point for users to interact with the grid, and handles processing tasks that are common to all grid services.

The Vega Grid "Browser" and the Vega Grid Server interact through a new protocol, called the Grid Service Request Protocol (GSRP). Another new feature is the Grid Service Markup Language (GSML), which allows users (not necessarily programmers) to specify grid services and user interface in an easy to use fashion.

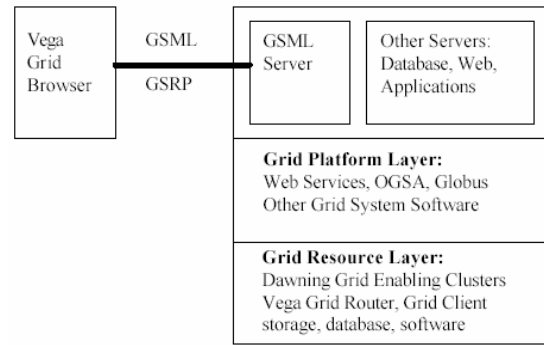


Figure 1. The Vega Grid Three-Layer Architecture

3. VEGA GRID BASED CONFIGURATION FOR SUNTHETIC FORCES EXPRESS

3.1 Task disassembling

SF Express simulation is based on DIS (Distributed Interactive Simulations). DIS is a network-based simulation environment intended for group training exercises, particularly military training. These simulations can be very complex, taking into account different vehicle types and behaviors, weather conditions, smoke and fires, local terrain, and surface soil types.

Each supercomputer communicates others by transmitting PDU(Protocol Data Units). With the increasing environment complexity and entity amounts result in a wealth of costs for communications, computational ability of parallel computers will be soon exhausted if there are no special dealing methods provided.

Fortunately, communications among entities have their own localization. Each supercomputer only need to calculate the relative components. The architecture based on Vega Grid as following can resolve the problems while expanding system dimensions.

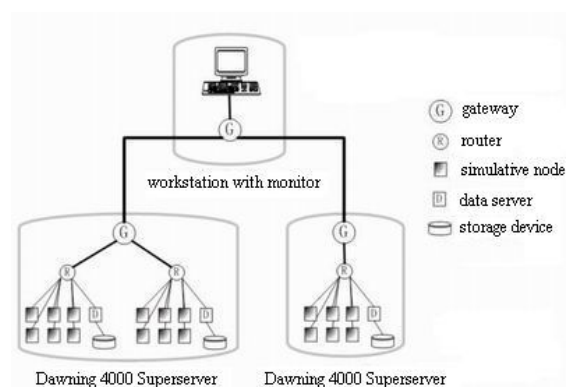


Figure 2. Vega Based SF Express Architecture

Description of primary node's function:

• **Simulator** Each simulator is a Grid processor to run the normal simulation program named ModSAFT, a popular DIS application program for military simulation. The Grid processor is a logical concept dissimilar to the traditional CPU. In practice, Grid processors can be personal computers, special client devices or software at client side. In the Vega

PG, the Grid processor mainly works as a control center, which manages the activities of Grid virtual hardware. Grid virtual devices provide only basic services and users can use the services to custom their activities freely. The conception of the Grid processor embodies the important philosophy of user centered but not server-centered.

- **Router** The router transmit messages for a group of simulators. It is a bridge linking all Grid processors and Grid virtual devices together. All Grid processors and Grid virtual devices can dynamically join or leave the Grid by connecting or disconnecting to routers. When more routers linked together, the Grid can expand to huge scale. The Grid can be constructed as a fully distributed resource network via routers. The router is a transfer station for message request, and it can collect the information of Grid virtual devices and give a path for message requests from Grid processors to Grid virtual devices^[13].

- **Gateway** The gateway is a special router with responsibility for message transmitting between local router and other parallel supercomputers, here refer to Dawning 4000 Superservers or further generations.

- **Data Server** The data server specially provides data store and access service to simulators, which includes large capacity store device and cache device.

The distribution topology of the virtual hardware is illustrated in Figure 3.

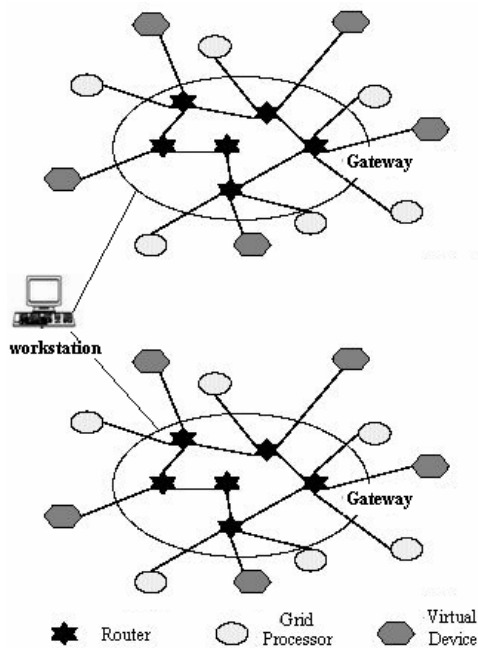


Figure 3. The topology of the Grid virtual hardware

In a Vega Grid environment for SF Express, we think the most important components are resource consumers and resource producers, which are represented by Grid processors and Grid virtual devices. Another important role in Grid virtual hardware is the Gateway that links all Grid processors and Grid virtual devices in a Dawning 4000 superserver together,

and it can transfer messages among parallel supercomputers.

To SF Express, the “base unit” is the simulator for the base calculation. The “middle unit” is the router node receives messages from each simulator and transmits them to the “high unit”. The “high unit” is the gateway node communicates among each Dawning 4000 Superserver. SF Express uses this three-layer architecture for message communication, and the purpose of using standardised PDU format is to simplify the information interaction of processes.

3.2 Syncrctizing with Globus

Why do SF Express needs the support of Globus? For the got-up decomposition of a task is steadfast. If there is merely a malfunction in one computer, the whole task can't be go on, we only can hope this case not happening. And the automatic collecting of SF Express running result or data log is only a fantasy, no to mention the dynamic analysis or adjusting computer's running status.

Maybe SF Express has possibility to succeed without the support of Globus, but it lacks elasticity toward the changing circumstance. Fortunately, Globus is a flexible middleware set which can help SF Express overcome the problems.

Globus provides several valuable capabilities to SF-Express. The Generic Resource Broker, Globus Resource Allocation Manager, and Metacomputing Discovery Service are used to initiate an SF-Express computation. Previously, SF-Express startup was a painful manual process. With Globus, the computation can be started from a single point.

The Globus Heartbeat Monitor provides a wide area mechanism for tracking the state of an SF-Express computation. Without the heartbeat monitor, one had to examine the contents of the SF-Express log file to ensure that a component of the computation had not failed.

The instance of syncrctizing between SF Express and Globus is showed by Figure 4.

For current environment, Each Dawning 4000 superserver installs GRAM(Globus Resource Allocation Manager) for reporting resource dynamic status to MDS(Metacomputing Directory Service), and DUROC(Dynamically -Updated Request Online Coallocator) is specially provided for dynamically updating request and communicating with GRAM

GEM(Globus Execution Management) is used to transmit executive code and initial data to each Dawning 4000 superserver and config initial status. So the application program can auto-adjust running status under the changing of environment.

The output data and running log produced in simulation process will auto-store to the concerning Dawning 4000 superserver by GASS(Global Access to Secondary Storage) for the purpose of real time monitoring, adjusting simulation status and display simulative war scene.

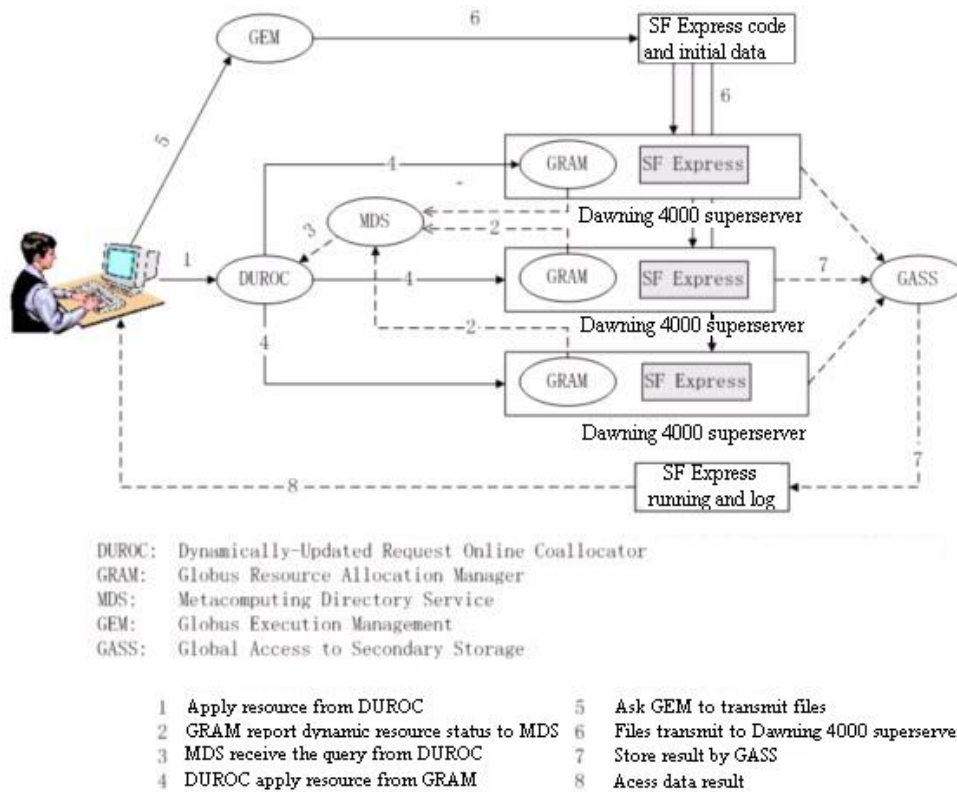


Figure 4. Synchronizing between SF Express and Globus

SF Express can't adapt environmental changing without the support of Globus. Globus is vital for boosting up the practicability of SF Express. Much energy should be pay to developing Globus software kits.

4. CONCLUSION

In this paper we propose a Vega Grid Based Configuration for Synthetic Forces Express. The architecture maps different components of the Grid to different components of a traditional computer system for SF Express. This Configuration helps us to construct the Grid environment for national military simulation. We also emphasize the importance of synchronizing between SF Express and Globus. In the next step, we will implement the unfinished functions of the Vega Grid for SF Express. The concerning middleware technology is the focus we should develop for further work. Another important work is consolidating the various Grid systems, such as OGSA, Web Service and Globus.

5. REFERENCES

- [1] SF-Express Application, <http://www.globus.org/research/applications/sfexpress.html>
- [2] The Globus Project, <http://www.globus.org/>
- [3] I. Foster, C. Kesselman, J. Nick, S. Tuecke, "Grid Services for Distributed Systems Integration", IEEE Computer, 35 (6), 2002, pp. 37-46.
- [4] I. Foster, C. Kesselman (Eds), The Grid: Blueprint for a New Computing Infrastructure, MorganKaufmann Publishers, 1998.
- [5] I. Foster, "What is the Grid: A Three-Point Checklist", Grid

Today, 1(6), 2002

- [6] I. Foster, C. Kesselman (Eds), the Grid: Blueprint for a New Computing Infrastructure, MorganKaufmann Publishers, 1998.
- [7] I. Foster, C. Kesselman, S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International Journal of Super-computer Applications, 15(3), 2001, pp. 200-222.
- [8] N.H. Sun, T.Y. Liu, "Grid Enabling Clusters", Journal of Computer Research and Development, 39 (8), 2002, pp. 917-922.
- [9] Z. Xu, W. Li, "The Research on Vega Grid Architecture", Journal of Computer Research and Development, 39 (8), 2002, pp. 923-929.
- [10] Z. Xu, W. Li, H. Fu, Z. Zeng, "The Vega Grid and Grid-Based Education", the 1st International Conference on Web-Based Learning, HK, China, 2002, 228-240.
- [11] Z. Xu, X. Li, G. Mei, "The Research on Architecture of Vega Information Grid", Journal of Computer Research and Development, 39 (8), 2002, pp. 948-951.
- [12] H. Zhuge, "A Knowledge Grid Model and Platform for Global Knowledge Sharing", Expert Systems with Applications, 22 (4), 2002, pp. 313-320.
- [13] W. Li, Z. Xu, F. Dong, J. Zhang, "Grid Resource Discovery Based on a Routing-Transferring Model", The 3rd International Workshop on Grid Computing, November, 2002.



Cheng Zhou, male, born in 1980.3, is the master of computer science and technology, Wuhan university of technology. His current research interests include parallel computing and grid computing.

Implementing Distributed Simulations in Grid Computing Environments*

Tingxin Song¹, Cheng Wang¹, Jianmin Xiong², Yaohe Liu²

¹ College of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan, Hubei Province, 430074, China;

² School of Mechanical Engineering, Hubei Polytechnic Universities, Wuhan, Hubei Province, 430064, China

Email: songtx@public.wh.hb.cn Tel.: +86 (0)27 88032313

ABSTRACT

Grid Infrastructure and popular Grid computing toolkit Globus Toolkit 3.0 (GT3) and SUN Grid Engine are introduced in this paper. A prototype of advanced distributed simulation system based on Grid computing is developed, working mechanism and programming model of this system is also discussed in this paper. Simultaneously, a three-dimension simulation experimentation oriented Grid computing about two collided black holes is described in this paper. The result indicates that computing capacity can be greatly improved by using Grid computing in advanced distributed simulation system.

Keywords: Grid, Grid Computing, Advanced Distributed Simulation, Globus

1 INTRODUCTION

It is difficult for using supercomputing to solve some problems in advanced distributed simulation system, for example, great capacity data located in different geographical area needs to handle, but Input/Output devices and high performance computing infrastructure are not located in local area. By Grid computing, various resources in network, such as supercomputers, great capacity of store facility, personal computers and all kinds of outer devices, are organized under a uniform framework. So Grid computing is a convenient approach to solve a distributed application problem. With evolvement of network technology, many supercomputers can be communicated on Internet, users can access computing power and hardware resources of super computers by network. Because of rapid development of Internet, Grid computing has become to the third Internet tidal wave followed after traditional Internet and Web. Traditional Internet realizes communication among computers hardware, Web realizes joining of web pages, Grid attempts to implement full connection to all resources on Internet. Grid computing means that high dependable and consistent hardware and software resources are available to different geographical users and resources. In this environment, many scientific research and engineering computing will be promoted greatly. Grid computing has become a crucial technology in information technology area^{[1][2]}.

Advanced distributed simulation system (ADS) is just a kind of system that can utilize Grid fully. ADS's essential is that simulation system's main nodes distributed in a widespread area are connected by LAN or WAN, ADS assigns computing works needed by large scope simulation environments to these main nodes. Grid computing being applied in ADS aims to

enable researches to leave laboratories to visit, monitor or administrate distributed experimentation platforms, and undertake collaborative examinations and remote computing by distributed Grid system.

2 BACKGROUND

2.1 Grid Operating System

At present, popular Grid operating system consists of task-centralized administration system, such as Sun Grid Engine、LSF(Load Sharing Facility)、PBS(Portable Batch System) and so on, and distributed task administration system, such as Globus Toolkit, Legion, NetSolve and so on. Tasks are scheduled by a computer in centralized system, while tasks in distributed system are performed and controlled by each computer in Grid system^[3].

2.2 Globus Toolkit 3.0 (GT3)

The open source project Globus Toolkit 3.0 is the most famous Grid system nowadays, it is a fundamental enabling technology for the "Grid", letting people share computing power, databases, and other tools securely online across corporate, institutional, and geographic boundaries without sacrificing local autonomy. The toolkit contains software services and libraries for resource monitoring, discovery, and management, plus security and file management. It is packaged as a set of components that can be used either independently or together to develop applications. The Globus Toolkit was conceived to remove obstacles that prevent seamless collaboration. Its core services, interfaces and protocols allow users to access remote resources as if they were located within their own machine room while simultaneously preserving local control over who can use resources and when Fig.1 illustrates the components on the server side. As shown in Fig.1, the major architecture components of the server side frameworks include the followings:

- The Web services engine. This engine is provided by Apache AXIS framework software and is used to deal with normal Web services behaviors, SOAP message processing, JAX-RPC handlers processing, and Web services configuration.

- Globus container framework. GT3 provides a container to manage the stateful Web service through a unique instance handle, instance repository, and life cycle management including service activation/passivation and soft-state management.

Currently GT3 uses Apache AXIS as its Web services engine,

* This work was supported by the National Defence Pre-research Foundation under Grant 413040402.

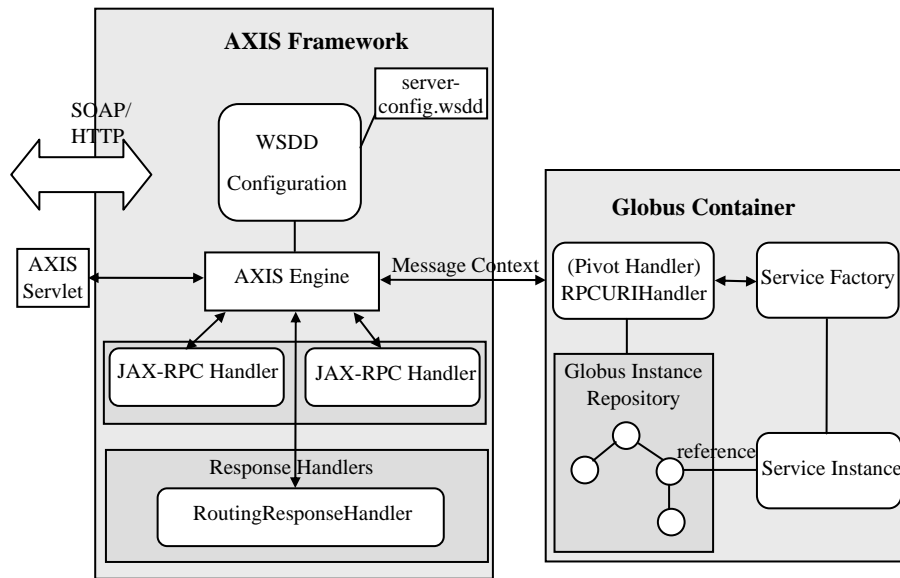


Fig.1 GT3 Software Framework: Server-side Architecture Components

which runs in a J2EE Web container and provides a SOAP message listener (AXIS servlet). It is responsible for SOAP request/response serialization and deserialization, JAX-RPC handler invocation, and Grid service configuration. As shown in Fig.1, GT3 container provides a pivot handler to the AXIS framework to pass the request messages to the Globus container.

This container architecture is used to manage the stateful nature of Web services and their life cycles. Once the service factory creates a Grid service instance, the framework creates a unique

Grid service handle (GSH) for that instance, and that instance is registered with the container repository.

Fig.2 illustrates the components of the client side.

As shown in Fig.2, Globus uses the normal JAX-RPC client-side programming model and AXIS client-side framework on Grid service clients. In addition to the normal JAX-RPC programming model, Globus provides a number of helper classes at the client side to hide the details of the OGSi client-side programming model^{[4]-[19]}.

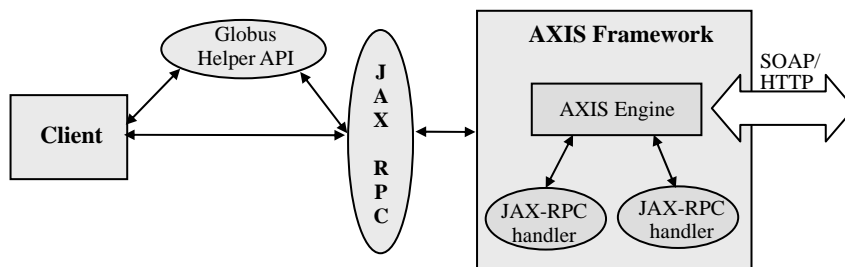


Fig.2 GT3 Software Framework: Client-side Architecture Components

2.3 Sun Grid Engines 5.3

Sun Grid Engine software was integrated with Globus toolkit since 1999, but it focus on business application. Sun Grid Engine software is a distributed management product that optimizes utilization of software and hardware resources. Sun Grid Engine finds a pool of idle resources and harnesses it productively, so an organization gets as much as five to ten times the usable power out of systems on the network. Sun Grid Engine software aggregates available compute resources and delivers compute power as a network service. Sun's Grid computing technology includes cluster Grid, enterprise Grid and global Grid. Sun Microsystems release Sun Grid Engine 5.3 Enterprise Version at June 2002, extending the idea of enterprise Grid architecture, promoting Grid computing

technology to a new phase, that is Enterprise Grid phase. Sun provides Sun Grid Engine 5.3 software (based Solaris and Linux platform) to Internet for download freely, playing an important role in Grid computing spreading worldwide^[10].

3 A PROTOTYPE OF ADVANCED DISTRIBUTED SIMULATION SYSTEM BASED ON GRID COMPUTING

3.1 System Architecture

We develop a prototype of advanced distributed simulation system based on Grid computing (ADSG), its architecture is shown as Fig.3.

Web pages shown in uniform interface are provided to users as service in this system, so ensuring the system can run on many different kinds of platform. So long as you can browse web pages, you will be able to use all resources provided by ADSG.

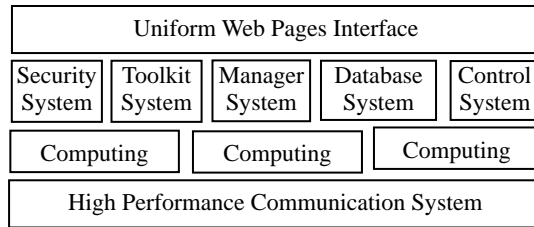


Fig.3 Architecture of ADSG

It can avoid users accessing computing system directly, avoiding conscious or unconscious destroying or attacking this system, and it is helpful to building uniform validated authorization mechanism. It is easy to use for most users, manipulation method based web is easy to be mastered, and it is helpful to spread high performance computing. Software system implements various kinds of controlling and administration function, downwards controls and managers various kinds of computing system, and upwards provides various kinds of service and realizes various kinds of computing task raised by users. Computing system is the provider of computing capability in ADSG, it contains various kinds of hetero-architecture, homo-architecture and different capability of computing systems which are consist of cluster, massively parallel processor system, or high performance server. ADSG is different from other computing system, it provides a uniform web interface and middleware over computing system, it is just middleware that transforms computing resource into web's computing capability, users can use computing facility in different geographical area and mask different computing system by web pages. So computing capability is "link and play" as electric power^{[11]-[13]}.

3.2 System Hardware Configuration

Four types of hosts are fundamental to the system. They are master host, execution host, administration host and submit host. Fig.4 displays the most important components and their interaction in the system.

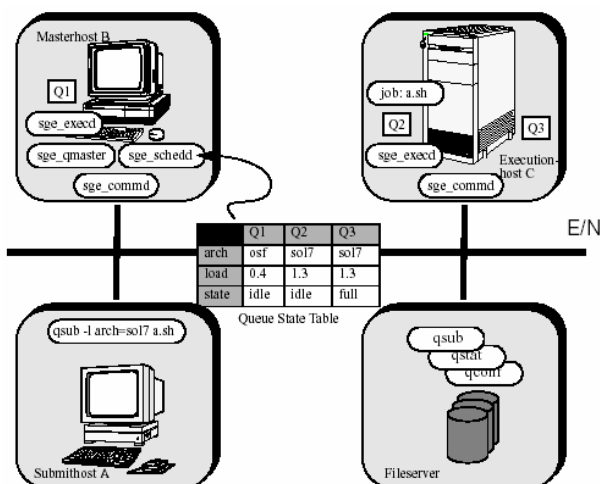


Fig.4 System Hardware Configuration

The master host is central for the overall cluster activity. It runs the master daemon and the scheduler daemon. Both daemons control all Grid Engine components, such as queues and jobs, and maintain tables about the status of the components, about user access permissions, and the like. By default, the master host is also an administration host and submit host.

Execution hosts are nodes that have permission to execute Grid Engine jobs. Therefore, they are hosting Grid Engine queues and run the Grid Engine execution daemon.

In Administration Host, permission can be given to hosts to carry out any kind of administrative activity for the Grid Engine system.

Submit hosts allow for submitting and controlling batch jobs only. In particular, a user who is logged into a submit host can submit jobs, and control the job status^[14].

4 SIMULATION EXPERIMENT BASED ON GRID COMPUTING

Based on above ADSG system and Cactus computing toolbox, we take a three dimension simulation experimentation regarding two collided black hole. The system architecture is shown as Fig.5. Executing process is as the following:

- (1) Start up Data Server;
- (2) Event: data available;
- (3) Start up Cactus computing;
- (4) Cactus reads data;
- (5) Cactus writes data;
- (6) Event: Cactus done;
- (7) Start up visualization;
- (8) Visualization reads data;

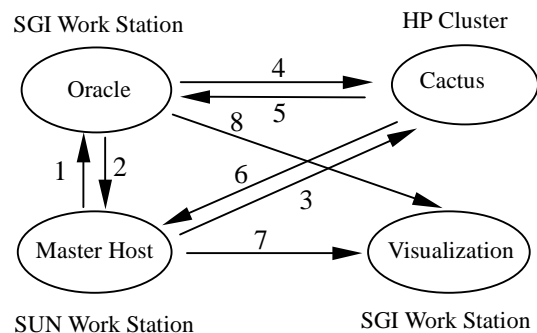


Fig.5 System Architecture of Black Hole Simulation

Cactus computing toolbox is an open source problem solving environment designed for scientists and engineers. The modular structure facilitates parallel computing across different architectures and collaborative code development between different groups. The Cactus code originated in the academic research community, where it has been developed and used over many years by a large international collaboration of physicists and computational scientists.

Executing code gets float point computing capacity of 66×10^9 times per second from HP minicom cluster, and the code can also run on SGI Origin2000 System high efficiently. In addition, Cactus can also run in a distributing environment in which there are 1500 SGI and IBM processors. Cactus toolbox can be

configured for remote simulation and manipulation. The code computes and simulates gravity wave radiated from black hole. Then visualization data is handled in parallel, and sent to a remote desktop by high speed network. Users may select and set simulating and visualization parameter remotely when the code is running. Globus Toolkit plays a very important role in Cactus which contains a web portal site on which users can configure and submit Cactus computing by web browser. With the help of message sending interface in Globus Toolkit, Cactus implements several distributed executing program spanning different platform^[15]. The last simulation results are shown as Fig.6.

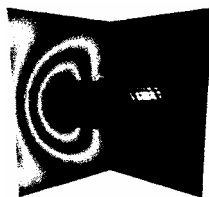


Fig.6 Black Hole Simulation

5 CONCLUSIONS

The current Grid computing system has been applied successfully in many domains, but Grid's structure should be more standardized for meeting the need of various kinds of sophisticated computing. It is getting started that Grid computing is applied in a factual distributed simulation system, there are many problems, such as system architecture, frameworks, development tool, Grid security and so on, needing to solved. In general, current Grid systems can only share resources in their own system, Connection among multiple Grid systems does not be implemented yet. Grid's spanning is a very important research focus for implementing a real seamless computing in network environment. It is predictable that as the latest technology in information technology, Grid computing being applied in system simulation technology, will extremely improve and facilitate the evolution of advanced distributed simulation system.

6 REFERENCES

- [1] Ian Foster, Carl Kesselman, Steven Tuecke. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Intl.J.Supercomputer Applications*. 15(3): p.200-222.
- [2] Ian Foster, Carl Kesselman, Jeffrey M. Nick, Steven Tuecke. *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, 2002. <http://www.globus.org/research/papers/ogsa.pdf>.
- [3] The OGSi WG and Specification, <http://www.gridforum.org/ogsi-wg/>.
- [4] Joshy Joseph, Globus ToolKit 3.0 and OGSi architecture overview. <http://www-106.ibm.com/developerworks/grid/library/gr-gt3/index.html>.
- [5] <http://www.globus.org/>.
- [6] <http://www.cactuscode.org/>.
- [7] <http://www-900.ibm.com/developerWorks/grid>.
- [8] <http://www.nasa.gov/>.
- [9] The OGSa Architecture documents, <http://www.ggf.org/ogsa-wg/>.

- [10] <http://www.sun.com/grid/>.
- [11] Enver Yucesan, Distribute web-based simulation experiments for optimization, *Simulation Practice and Theory* 9(2001) 73-90.
- [12] U.Klein. Simulation-based distributed systems: serving multiple purposes through composition of components, *Safety Science* 35 (2000) 29-39. <http://www.elsevier.com/locate/ssci>.
- [13] S.P.Murphy, T.Perera. Successes and failures in UK/US development of simulation, *Simulation Practice and Theory* 9 (2002) 333 - 348. <http://www.elsevier.com/locate/simpra>
- [14] S. Brunett, O. Davis T.gottschalk, P. Messina. Implementing Distributed Synthetic force simulation in meta-computings environments. <http://www.globus.org/about/>
- [15] K. Keahey, T. Fredian, Q. Peng, D. P. Schissel, M. Thompson, I. Foster. *Computational Grids in Action: The National Fusion Collaboratory*, 2001. The National Fusion Collaboratory. <http://www.fusiongrid.org>



Tingxin Song is a doctor candidate in College of Hydro- power and Information Engineering, Huazhong University of Science and Technology, he is a teacher as well in Hubei Polytechnic University. He graduated from Hubei Polytechnic University in 1997 with specialty of Mechanical Engineering. His main academic interesting on system simulation technology, grid computing and computer measure and control.

HACP: An Ant-Based Partitioner for Grid Computing Applications*

Lin Jin¹, Wang Meiqing², Jiang Xiufeng³

College of Mathematics and Computer Science, Fuzhou University

Fuzhou, Fujian 350002, China

Email: ¹ljin1997@hotmail.com ²mqwang@fzu.edu.cn ³jxf1963@hotmail.com Tel: 0591-3795095

ABSTRACT

Efficient use of the distributed resources (computers, databases, and human expertise, etc.) requires that the computational load must be balanced across processors in a way that minimizes communications among processors and data moving. Multilevel partitioners such as Minex have been used for grid computing application in recent research works. In this paper, we present a novel map algorithm called HACP, an ACO algorithm coupled with a local search. ACO algorithm, due to their intrinsically distributed and multi-agent nature that well matches these types of architectures, can be very effective. To efficiently solve computational load-balance problems we give a parallel implement model for HACP. Experimental results demonstrate that HACP is much more efficient than the partitioning method in MinEx.

Keywords: Information Power Grid, Partitioning, Ant Colony Optimization, local search

1. INTRODUCTION

NASA and its collaborative partners are actively developing the Information Power Grid (IPG) to harness the vast collection of their geographically distributed resources (computers, databases, and human expertise). One of the primary benefits of the IPG will be to facilitate the efficient solution of large-scale computational problems by providing a scalable, adaptive, and transparent environment that is both ubiquitous and uniformly accessible through a convenient interface. The IPG is one of several approaches to develop what are called computational Grid (in short, grid) capabilities and /or implementations. At the present, only a few limited studies have been performed at NASA Ames Research Center to determine the viability of large-scale parallel and distributed computing on the IPG. For grid computing application, the architecture of a computer network and its poor connectivity and large latencies due to geographical separation in a realistic IPG environment could impact overall performance.

The MinEx [1], a latency-tolerant dynamic partitioner for grid computing application has proposed in paper [1]. MinEx optimizes computation, communication, and data remapping costs. It redefines the partitioning goal from producing balanced workloads to minimize the Max processing cost among the processors. MinEx can be classified as a diffusive multilevel partitioner. Diffusive algorithms [2] utilize an existing partition as a starting point instead of partitioning from scratch. It partitions a graph in three steps: contraction, partitioning, and expansion. It is said that once the mesh is sufficiently contracted, the remaining vertices are reassigned according to the partitioning criteria described in section three.

Then the mesh is expanded back to its original size through a refinement process.

The partition scheme in MinEx is a greedy-like algorithm. We find the performance is poor when applying it in some computational case. In this paper, we propose a novel algorithm, called Hybrid Ant Colony Partitioner (HACP), which do better than the MinEx partitioner when reassigning the vertexes to the processors.

In fact, as in the case of parallel and/or network processing, the computational architecture is spatially distributed, ACO algorithms, due to their intrinsically distributed and multi-agent nature that well matches these types of architectures, can be very effective [3]. Combining a good local search we apply the ACO algorithm to the grid computing application and let the agents work parallel. Experimental results demonstrate that HACP is much more efficient than the partitioning method in MinEx.

The rest of this paper is organized as follows. Section 2 describes MinEx, including the various graphs and the metrics that model the computational application and the multilevel partitioner. Section 3 introduces the ACO algorithm. Section 4 presents our new partitioner HACP. Finally we give computational results and conclusions.

2. MINEX

In this section we describe the computational model and the multilevel partitioner in MinEx.

2.1 Computational model

We discretely model computational problems as an unstructured mesh $G_1 = (V, E)$, where V is the set of vertexes that represent the tasks in grid applications, E is the set of edges that connect those vertexes. Each vertex v has two weights, $Pwgt_v$ and $Rwgt_v$, while each edge (v, w) has one weight $Cwgt_{(v,w)}$. These weights respectively model the associated computational processing, data remapping, and runtime inter-processor communication costs.

To predict performance on a variety of distributed architectures, a configuration graph G_2 is utilized. Each vertex in this fully connected graph represents a tightly coupled cluster of processors, while edges denote cluster interconnections. For the sake of the experiments presented in this paper, we assume that all processors in a cluster are homogeneous and that there is a constant bandwidth for intra-cluster communication. Each vertex in the configuration graph has an associated weight $Proc_c \geq 1$ representing the processing slowdown factor for a cluster relative to the others. Likewise, the weight of an edge $Conn_{(c,d)} \geq 1$ represents the

* This work was supported by the Educational Ministry of Fujian Province, China under Grant No. JA02147.

interconnect slowdown factor when a processor in cluster c communicates with a processor in cluster d . If $c = d$, $Conn_{(c,c)}$ represents the slowdown associated with communication between processors in the same cluster c . Of course if two vertex are in the same processor then $Conn_{(c,c)} = 0$.

The following metrics respectively measure the number of time units required for computation, data remapping, and communication. The total time required to process the vertices assigned to a processor p must take into account all three metrics:

- Processing cost: Wgt_v^p is the computational cost to process the vertex v assigned to processor p which is in cluster c :

$$Wgt_v^p = Pwgt_v * Proc_c$$

- Communication cost: $Comm_v^p$ is the communication cost to interacted with all vertex w adjacent to v whose data sets are not local to p (assuming that v is assigned to p):

$$Comm_v^p = \sum_w Cwgt_{(v,w)} * Conn_{(c,d)}$$

Where c and d are the clusters containing the processors to which v and w are respectively assigned.

- Redistribution cost: $Re map_v^p$ is the overhead to copy the data set associated with v to another processor from p :

$$Re map_v^p = Rwgt_v * Conn_{(c,d)}$$

- Weighted queue length: $Qwgt^p$ is the total cost to process all vertex v assigned to p :

$$Qwgt^p = \sum_v (Wgt_v^p + Comm_v^p + Re map_v^p)$$

- Total system load: $QwgtTot$ is the cost to process the entire application:

$$QwgtTot = \sum_p Qwgt^p$$

- Heaviest load: $MaxQwgt$ indicates the total time required to process the whole application:

$$MaxQwgt = \max_p Qwgt^p$$

- Lightest load: $MinQwgt$ indicate the workload of the most lightly loaded processor:

$$MinQwgt = \min_p Qwgt^p$$

- Load imbalance factor: $Load Imb$ represents the quality of the partitioning:

$$Load Imb = P * \frac{MaxQwgt}{QwgtTot}$$

Where P is the total number of processor in the configuration graph.

2.2 Multilevel partitioner

Similar to other multilevel partitioners, the first step in MinEx is to contract the mesh to a reasonable size. MinEx randomly selects a set of adjacent vertex pairs that are assigned to a same processor. From this set, the vertex pair (v, w) that has

the largest value $Cwgt_{(v,w)} / (Rwgt_v + Rwgt_w)$ is merged.

This formula attempts to find edges with large communication costs while minimizing the potential data redistribution overhead. The motivation behind this strategy is to arrive at a contracted mesh with a small edge cut as well as a small data distribution cost.

The partitioning is performed when the graph contraction process is complete. For each vertex v , consider all adjacent vertices assigned to other processors. Compute the *Gain* and *MinVar* values, where the *Gain* represents the change in $QwgtTot$ that would result from a proposed vertex motions; the *MinVar* measures the variance of processor workloads from that of the most lightly loaded processor. Both values would result from moving v to each of these adjacent processors. The vertex moved is the one with the smallest *Gain* and satisfies $\Delta MinVar < 0$ and $-Gain / \Delta MinVar < ThroTtle$, where the *ThroTtle* is a user-supplied parameter; It is computed using the workload for each processor p and the smallest load over all processors:

$$MinVar = \sum_p (Qwgt^p - MinQwgt)^2$$

$\Delta MinVar$ is the change in *MinVar* after moving a vertex from one processor to another.

The partitioning algorithm favors vertex motions with negative $\Delta MinVar$ or small *Gain* values that reduce or minimize the overall system load, and tend to move vertices away from processors that have high runtime requirements. *ThroTtle* acts as a gate that limits increases in *Gain* based upon how much of an improvement in *MinVar* can be achieved.

Finally, the mesh is expanded back to its original size through a refinement process. As each vertex is reinstated, a decision is made as to whether or not it should be reassigned.

3. THE ACO ALGORITHM

Ant algorithm [4] is a class of population-based meta-heuristic algorithm, is suitable for solving hard combinatorial optimization problems.

A colony of social insects, such as ants, bees, usually performs its own tasks independently from other members of the colony. However, the tasks performed by different insects are related to each other in such a way that the colony, as a whole, is capable of solving complex problems through cooperation. Ethologists discover that a colony of ants are capable of finding the shortest path between a food source and the nest (adapting to changes in the environment) without the use of visual information. In order to exchange information about which path should be followed, ants communicate with one another by means of pheromone trails. As ants move, a certain amount of pheromone is dropped on the ground. The more ants follow a given trail, the more attractive this trail becomes to be followed by other ants. This process can be described as a loop of positive feedback, in which the probability that an ant chooses a path is proportional to numbers of ants that have already passed by that path.

When an established path between a food source and the ant nest is disturbed by the presence of an object, ants soon will try to go around the obstacle. Firstly, each ant can choose to go around to the left or to the right of the object with a 50%-50% probability distribution. All ants move roughly at the same speed and deposit pheromone in the trail at roughly the same rate. Therefore, the ants that (by chance) go around the obstacle by the shortest path will reach the original track faster than the others that have followed longer paths to circumvent the obstacle. As a result, pheromone accumulates faster in the shorter path around the obstacle. Since ants prefer to follow trails with larger amounts of pheromone, eventually all the ants converge to the shorter path. An Ant Colony Optimization algorithm (ACO) is essentially a system based on agents which simulate the natural behavior of ants, including mechanisms of cooperation and adaptation.

In essence, the design of an ACO algorithm involves the specification of following:

- An appropriate representation of the problem, which allows the ants to incrementally construct/modify solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local, problem-dependent heuristic.
- A rule for pheromone updating, which specifies how to modify the pheromone trail (τ).
- A probabilistic transition rule based on the value of the contents of the pheromone trail (τ) that is used to iteratively construct a solution.

4. HYBRID ANT COLONY SYSTEM FOR PARTITION

In this paper we design an ACO – a hybrid ant colony partition (HACP) to optimize multilevel partition problems in grid applications. According to the multilevel partitioning method, the mesh is contracted to a reasonable size; then we use the HACP to repartition the mesh G_1 that will minimize the $MaxQwgt$, the total time required to process the application. After optimizing current partition we expand the graph to original size.

To applying the ACO algorithm for applications, we represent a partition problem as a graph $G=(G_1, G_2, E)$, where G_1 is the unstructured mesh described in section two. The G_1 models the computing problem, and G_2 is a configuration graph which models the distributed architecture. Each vertex of G_2 represents a processor. Each vertex in G_1 has edges connecting to all vertexes in G_2 and E is the set of these edges. Each edge of E has a weight $\tau(i, j)$, which represents the amount of trail the ant deposited. We use the trail metric $\tau(i, j)_{n \times p}$ to represent the trail of all edges of E , where the elements in row i represent the trail of the edges between vertex i and vertexes in G_2 .

To begin the algorithm, initialize the trail of each edge as $mul / MaxQwgt$, where mul is a constant, for example, in the experiment we let $mul = 1000$, $MaxQwgt$ is the heaviest load of current partition.

Now we distribute n sub-colonies of agents in each vertex of G_1 , where n is the number of vertexes of G_1 , each sub-colony has P agents, where P is the number of the processors. The task of an agent is to select a processor for a vertex and the choice rule is a pseudo-random-proportional rule. The agent in vertex i choose process j to move according to following formula:

$$j = \begin{cases} \arg \max_{u \in \{0, 1, \dots, P-1\}} \tau(i, u) & \text{if } q \leq q_0 \\ j_1 & \text{others} \end{cases} \quad (1)$$

where $q_0 = 0.9$, $0 < q < 1$, q is a value chosen randomly with uniform probability, $\arg \max \tau(i, u)$ represents the processor u who connects with vertex i and the trail on the edge is the max among all edges connecting to vertex i . j_1 is a random variable selected according to a probability distribution :

$$P_k(i, j) = \frac{\tau(i, j)}{\sum_{\substack{s \in U \setminus \arg \max \\ u \in \{0, 1, \dots, P-1\}} \tau(i, u)}} \quad (2)$$

$$j \in U \setminus \arg \max_{s \in \{0, 1, \dots, P-1\}} \tau(i, s)$$

where U is the vertex set consisted of all vertex connecting with vertex i .

The formulas (1) and (2) mean that the edge with the max trail will be chosen with high probability 0.9, the others with $0.1 * p_k(i, j)$. But in a standard ant system, the formula is as follow:

$$p_k(i, j) = \frac{\tau(i, j)}{\sum_{s \in U} \tau(i, s)} \quad j \in U \quad (2')$$

According to formula (2') the edge with the max trail is still chosen with high probability, as a result, the agent build many similar solutions, the algorithm convergences rapidly. According to our choice rule (2), the edge with max trail is chosen with high probability 0.9, the other edge can always be selected with low probability, thus the agent can build different partitions in every iteration before exploitation of the max trail to avoid a too rapid convergence of the algorithm towards a sub-optimal region.

Having moved to the processor chosen, the agent decreases the trail of the edge properly so that the other agent will not select this edge with so high probability, which favor the exploration of new areas of the search space. This communicating action can't be implemented by a single agent, but a colony of agent.

The trail value is updated locally by the following rule:

$$\tau(i, j) = (1 - \alpha)\tau(i, j) + \alpha \min \tau_i \quad (3)$$

where $\alpha = 0.9$, $\min \tau_i$ is the minimal value in the row i of the trail matrix. So the update term $\tau(i, j)$ is composed mostly of the discounted trail but it will not less than the minimal

value in the row.

After those sub-colonies completed their selection independently, P partitions are built. A partition is a $1 \times N$ matrix, each element i of which represents the processor where the vertex i to be assigned. For example, the first agent of all sub-colonies builds a partition.

After a model is given, an improvement phase is implemented. We apply a fast neighborhood search called 3-opt to improve the solution rapidly. These partitions built by the colonies act as the initial values. The 3-opt algorithm is described below:

```

{Initialize  $Partition_0 = (x_1, x_2, \dots, x_n)$ ;
 $Num = 0$ ;
 $X = Partition_0$ ;
Do
{Select a vertex  $\bar{X}$  randomly in  $N(X)$ ;
Compute New_MaxQwgt for the application;
If (  $New\_MaxQwgt < MaxQwgt$  )
Then {  $X = \bar{X}$ ;
 $MaxQwgt = New\_MaxQwgt$ ;
 $Num = 0$ ; }
Else  $Num++$ ;
} While (  $Num < Max\_Num$  )
Where  $N(X) = \{ \text{all possible } X' \mid X' \text{ is obtained by randomly modify 3 element } x_i, x_j, x_k \text{ of } X \text{ to be a randomly value } x'_i, x'_j, x'_k, x'_i, x'_j, x'_k \in \{0, 1, \dots, P-1\} \}$ ,
and  $Max\_Num$  is a parameter defined by the user.
```

3-opt method works better than the partition method in MinEx which can be looked as 1-opt. Our goal of the 3-opt is not to find the local optimal value but to improve the solution rapidly. To reduce implementing time we stop the algorithm when the solution not improved after maximum times.

After the agents have modified the existing partition, the information contained in pheromone trail matrix is taken into account. The update of the pheromone trail is done here in a different way than those of the standard models where all the ants update the pheromone trail. Indeed, this method of updating the pheromone trail implies a very slow convergence of the algorithm. For speed-up the convergence, we update the pheromone trail by taking into account only the best solution produced by the search to date. First, all the pheromone trails are weakened by setting

$$\pi_{ij} = (1 - \alpha) * \tau_{ij}, (1 \leq i, j \leq n) \text{ where } 0 < \alpha < 1 \text{ is a}$$

parameter that controls the evaporation of the pheromone trail: a value of α close to 0 implies that the pheromone trails remains active a long time, while a value close to 1 implies a high degree of evaporation and a shorter memory of the system. Then, the trail is reinforced by considering only the best solution generated by the system so far. The global update rule is below:

$$\tau(i, j) = (1 - \alpha) \cdot \tau(i, j) + \alpha \cdot \Delta\tau(i, j)$$

where

$$\Delta\tau(i, j) = \begin{cases} \frac{mul * \sum_{k=0}^{P-1} \tau(i, k) - \tau(i, j)}{\sum_{k=0}^{P-1} \tau(i, k)} & \text{If } (i, j) \text{ is the edge of the best partition} \\ 0 & \text{Otherwis} \end{cases} \quad (4)$$

The formula (4) is different from the standard system too. It can be explained as that if the trail of the edge (i, j) is much larger than that of other edges, the ant doesn't need to deposit too much trail; while the trail is much little than others, the ant will deposit more trail so that the trail on the edge of the best solution the ant found randomly could be increased rapidly in next iteration. The pheromone information will lead the search of the future ants. They will search around the good partition.

The programming style used is a synchronous master/workers paradigm. The master implements a central memory through which passes all communication, and captures the global knowledge acquired during the search. The workers implement the search processes. The parallel Algorithm works as follows:

1. The master initialize the trail matrix, compute the MaxQwgt of the current partition, and spawn a set of processes, one for each vertex. Then it sends the corresponding trail information to the processors.
2. For $Num := 1$ To $Imax$ Do
 - 2.1 /* Each worker lets the ants select a processor sequentially then sends the selection to the master */
Parallel: {For (agent=0 to $n-1$)
 {The current agent chooses a processor for vertex i according to formula (1,2);
 Update the trail of the edge of (i, j) according to formula (3); }
 Send their solution to master; }
 - 2.2 The master receives the solution from the workers and generate p partitions, then sends the p partitions to p workers
 - 2.3 Parallel: { The workers starting from these partitions do a 3-opt local search to improve the solution ; then send their improved partitions f_0, f_1, \dots, f_{p-1} to the master. }
 - 2.4 The master receives the p partitions and update the trail using the best partition according to formula(5)
End_Do
- 3 Ouput: MaxQwgt and Partitions.

5. EXPERIMENTAL RESULTS

We compare the quality of the partitions produced by our HACP with those by the partitioning method in MinEx .In the experiments, we assume that a graph has been contracted sufficiently and the $Pwgt_v$ and $Cwgt_{(v,w)}$ are generated randomly. The different network topologies are also given, for example the number of processors、the number of clusters and the slowdown factors are set to different. The intra-cluster communication speeds are normalized to a unity. The inter-cluster slowdown factors are assumed as 3,10,100,1000. To simplify the analysis, we also assume that all processors are homogeneous and are divided as evenly as possible among the clusters.

The tables show the values of MaxQwgt and LoadImb of our experiments which runs at varying numbers of clusters, interconnect speeds and processors .

Table1 : MaxQwgt and LoadImb ,Processor=16

	c	Interconnect slowdowns			
		3	10	100	1000
MinEx	1	6241 (1.5)	6133 (1.6)	7499 (2.0)	6755 (1.7)
	2	6300 (1.6)	20017 (2.1)	126990 (1.6)	1227354 (1.9)
	4	6967 (2.5)	16540 (2.1)	180070 (2.4)	1791709 (2.6)
	8	7165 (2.8)	20121 (2.4)	187149 (2.8)	2024299 (3.3)
HACP	1	5226 (1.3)	4846 (1.3)	5042 (1.3)	5277 (1.3)
	2	4634 (1.3)	8783 (1.2)	75160 (1.3)	846945 (1.3)
	4	4524 (1.6)	12383 (1.6)	121341 (1.7)	1279891 (1.6)
	8	3521 (1.6)	9650 (1.5)	95458 (1.5)	887803 (1.6)

Table2 MaxQwgt (LoadImb), Slowdown factors=3

method	processor	MaxQwgt (LoadImb)	
		C=1	C=8
MinEX	8	6886(1.5)	5863(1.5)
	16	4778(2.0)	7165(2.8)
	32	2760(2.3)	3554(4.0)
	64	2265(3.7)	4595(4.0)
HACP	8	5110(1.1)	4361(1.2)
	16	2890(1.2)	3521(1.6)
	32	1715(1.4)	1741(2.1)
	64	1403(2.3)	2275(1.9)

The experimental results show that HACP can be robust in variable kinds of network topologies. It produced much better partitions than that of the MinEx. The partition method in MinEx can be considered as a 1-opt algorithm, which reassign one vertex each time; in addition, the reassigning processes only happen N times. The 3-opt algorithm outperforms than the 1-opt but it is easier to go to the local optimum than our HACP, because the HACP algorithm is a population-based algorithm. A good quality solution can emerge due to the result of the collective interaction among the agents.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a novel partitioner HACP, which is suitable for adaptive mesh applications executed in a parallel-distributed method on NASA's IPG due to their intrinsically distributed and multi-agent nature. We use the goal function defined in MinEx, apply the multilevel partition scheme, and present a parallel model for the ant colony to solve the assigning problem. Experimental results demonstrate that HACP is much more efficient than the partition method in MinEx. HACO can be very promising and robust.

Future work includes three important directions. First, real distributed experiments using Globus are planned to complement the results in this paper. Second improve the parallel model. Finally, investigate pheromone updating strategy and the heuristic function that has not been used in this paper.

7. REFERENCES

- [1] Sajal K. Das , Daniel J. Harvey , Rupak Biswas. "MinEX: a latency-tolerant dynamic partitioner for grid computing applications." *Future Generation Computer Systems* 18 (2002), 477-489
- [2] George Karypis and Vipin Kumar. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs" *SIAM Journal on Scientific Computing* Volume 20, Number1 p359-392 1998 Society for Industrial and Applied Mathematics.
- [3] Marco Dorigo and Gianni Di Caro. "The Ant Colony Optimization Meta-Heuristic" [Http://iridia.ulb.ac.be](http://iridia.ulb.ac.be)
- [4] M.Dorigo, A. Coloni and V. Maniezzo, "The Ant System: optimization by a colony of cooperating agents." *IEEE Transactions on Systems, Man, and Cybernetics-part B*, vol.26, no. 1, pp.29-41, 1996.
- [5] Luca M. Gambardella, Marco Dorigo. Ant-Q: A Reinforcement Learning Approach To the TravelingSalesman Problem. *Proceeding of ML-95, Twelfth Intern. Conf. on Machine Learning*, Morgan Kaufmann, 1995, 252-260.
- [6] L.M.gambardella, E.D. Taillard, M.Dorigo. Ant Colony for the QAP. *Technical Report IDSIA-4-97*, 1997.
- [7] Marco Dorigo, Luca Maria Gambardella. Ant colonies for the traveling salesman problem. *TR/IRIDIA/1996_3*.
- [8] Rafael S. Parpinelli, Heitor S. Lopes, and Alex A. Freitas. Data Mining with an Ant Colony Optimization Algorithm. *IEEE Trans on Evolutionary Computation*, Special issue on Ant Colony Algorithms, 6(4), August 2002.

A Service-oriented Grid Computing Model Based on Jini

Tang Guosheng, Guo Qingping, Yang Jian

School of Computer Science and Technology, Wuhan University of Technology

Wuhan, Hubei 430063, China

Email: gshtang@yahoo.com Tel: 027-86533605

ABSTRACT

Jini is a promising piece of Java technology designed to create dynamic distributed object system. This paper describes Jini architecture and introduces its component how to harmoniously work in a service-oriented distributed computing platform. Then, this paper builds a grid computing model by extending Jini, illustrates the basic framework of this model and states its implement mechanism.

Key words: Java, Jini, Distributed Computing, Service-oriented, Grid, Grid computing

1. INTRODUCTION

The Internet is changing the way we do many things. But most importantly, it is changing the concept of networks from multiple connected computers to connected devices, including computers, which deliver smart Web services through the network. Although Jini technology is not a Web service, it can communicate with, or be used to build, Web services. Jini technology was developed to address an important set of long-term problems that must be solved before generic Web services can be transformed into the highly personalized, cross-network, intelligent services that future systems will provide.

As the next level beyond network connectivity, Jini technology provides developers with tools to construct systems from distributed objects over networks. It offers a simple infrastructure for delivering services over the network and creating spontaneous interaction between programs that use these services, regardless of their hardware or software implementations.

Any type of network made up of services and clients of those services can be easily assembled, disassembled, and maintained on the network using Jini technology.

2. JINI ARCHITECTURE

A Jini system is a distributed system based on the idea of federating groups of users and the resources required by those users. The overall goal is to turn the network into a flexible, easily administered tool on which resources can be found by human and computational clients. Resources can be implemented as hardware devices, software programs, or a combination of the two. The focus of the system is to make the network a more dynamic entity that better reflects the dynamic nature of the workgroup by enabling the ability to add and delete services flexibly.

A Jini system consists of the following parts[2]:

- A set of components that provides an infrastructure for federating services in a distributed system

- A programming model that supports and encourages the production of reliable distributed services
- Services that can be made part of a federated Jini system and which offer functionality to any other member of the federation

While these pieces are separable and distinct, they are interrelated, which can blur the distinction in practice. The components that make up the Jini technology infrastructure make use of the Jini programming model; services that reside within the infrastructure also use that model; and the programming model is well supported by components in the infrastructure (Figure 1).

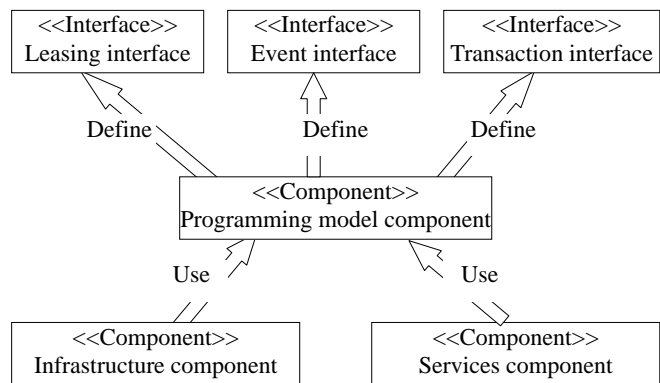


Fig.1 Jini component

A Jini system can be seen as a network extension of the infrastructure, programming model, and services that made Java technology successful in the single-machine case. These categories along with the corresponding components in the familiar Java application environment are shown in Figure 2.

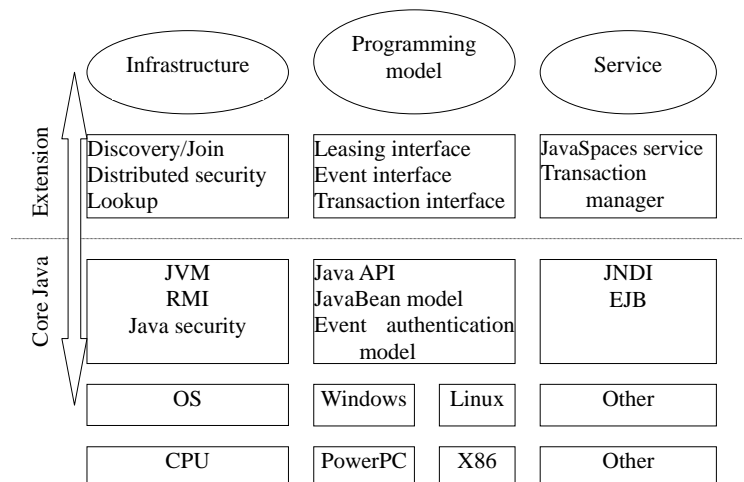


Fig.2 Java extension for Jini

The Java application environment provides a good computing platform for distributed computing because both code and data can move from machine to machine. The environment has built-in security that allows the confidence to run code downloaded from another machine. Strong typing in the Java application environment enables identifying the class of an object to be run on a virtual machine even when the object did not originate on that machine. The result is a system in which the network supports a fluid configuration of objects which can move from place to place as needed and can call any part of the network to perform operations.

The Jini architecture exploits these characteristics of the Java application environment to simplify the construction of a distributed system. The Jini architecture adds mechanisms that allow fluidity of all components in a distributed system, extending the easy movement of objects to the entire networked system.

Infrastructure

The Jini technology infrastructure defines the minimal Jini technology core.

The infrastructure includes the following:

- A distributed security system, integrated into RMI, which extends the Java platform's security model to the world of distributed systems
- The discovery/join protocol, a service protocol that allows services (both hardware and software) to discover, become part of, and advertise supplied services to the other members of the federation
- The lookup service, which serves as a repository of services. Entries in the lookup service are objects in the Java programming language; these objects can be downloaded as part of a lookup operation and act as local proxies to the service that placed the code into the lookup service

The discovery/join protocol defines the way a service of any kind becomes part of a Jini system; RMI defines the base language within which the Jini services communicate; the distributed security model and its implementation define how entities are identified and how they get the rights to perform actions on their own behalf and on the behalf of others; and the lookup service reflects the current members of the federation and acts as the central marketplace for offering and finding services by members of the federation. The mechanism which the infrastructure implements distributed computing is shown in Figure 3.

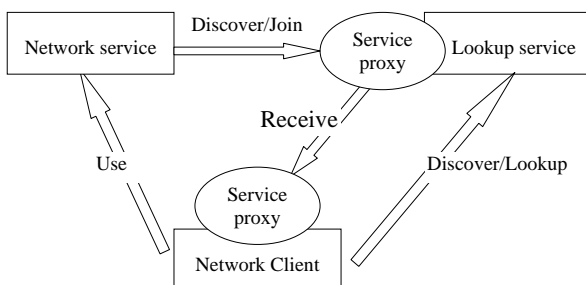


Fig.3 Jini implementation mechanism

Programming Model

Both the infrastructure and the services that use that infrastructure are computational entities that exist in the physical environment of the Jini system. However, services also constitute a set of interfaces that define communication

protocols that can be used by the services and the infrastructure to communicate between themselves.

These interfaces, taken together, make up the distributed extension of the standard Java programming language model that constitutes the Jini programming model. Among the interfaces that make up the Jini programming model are the following:

- The leasing interface, which defines a way of allocating and freeing resources using a renewable, duration-based model. It extends the Java programming language model by adding time to the notion of holding a reference to a resource, enabling references to be reclaimed safely in the face of network failures.
- The event and notification interface, which is an extension of the event model used by Java Beans components to the distributed environment that enables event-based communication between Jini services. They extend the standard event models used by JavaBeans components and the Java application environment to the distributed case, enabling events to be handled by third-party objects while making various delivery and timeliness guarantees. The model also recognizes that the delivery of a distributed notification may be delayed.
- The transaction interfaces, which enable entities to cooperate in such a way that either all of the changes made to the group occur atomically or none of them occur. They introduce a lightweight, object-oriented protocol enabling Jini applications to coordinate state changes. The transaction protocol provides two steps to coordinate the actions of a group of distributed objects.

Services

The Jini technology infrastructure and programming model are built to enable services to be offered and found in the network federation. These services make use of the infrastructure to make calls to each other, to discover each other, and to announce their presence to other services and users.

Services appear programmatically as objects written in the Java programming language, perhaps made up of other objects. A service has an interface that defines the operations that can be requested of that service. Some of these interfaces are intended to be used by programs, while others are intended to be run by the receiver so that the service can interact with a user. The type of the service determines the interfaces that make up that service and also define the set of methods that can be used to access the service. A single service may be implemented by using other services.

The Jini services include the following:

- A JavaSpaces service, which can be used for simple communication and for storage of related groups of objects written in the Java programming language
- A transaction manager, which enables groups of objects to participate in the Jini Transaction protocol defined by the programming model

3. A GRID COMPUTING MODEL EXTENDED JINI

The early period of grid computing viewed grids as a collection of resources: processors, memory, instruments, and so on. Today, grids are universally seen as a collection of

services. A service is an abstract concept, representing a piece of computation, an application or a component, or a hardware device, such as a stereo display, processor, or a measurement instrument. The services that make up a grid can be considered virtualized resources.

For a client, whether human or another software component, to access a service, that service must be described by a public interface. A service's public interface defines a contract with that service's clients, allowing clients to rely only on that contract to interact with a service. Distributed systems built around services, or virtualized resources, with services having clearly defined interfaces, are often termed service-oriented systems.

As a service-oriented distributed computing system based on the Java programming language, Jini naturally matches the requirements of grids composed of services. Participants in a Jini network community are clients and services. Every Jini service is described by a Java programming language interface. Since Java supports a strong type system, relying on Java interfaces for service definition allows Jini services in a grid to adhere to strong type definitions as well, and enables type-based service discovery.

Grid Computing Model

We can extend some important Jini features for building a grid computing model (Figure 4): spontaneous service discovery, service proxies, dynamically downloadable user interfaces, lease-based access to resources, and security.

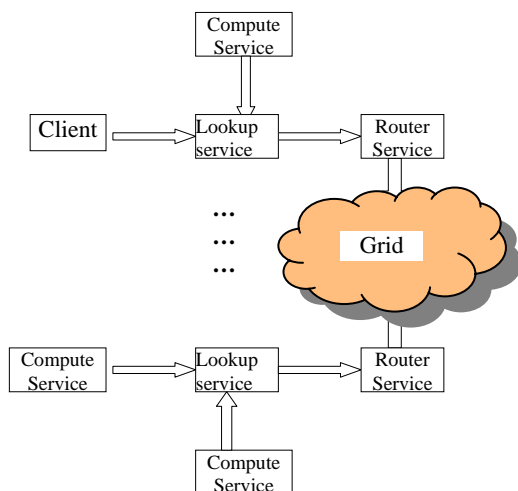


Fig.4 A Grid Computing Model

Compute Service

Firstly we need to extend network service for compute service so that it can supply client with compute service. In this model, Compute services are the key components. The compute service has the role of executing Java objects originated from a remote client. They export Java Virtual Machines of remote computers into the Jini community and their role is to allow user created Java objects to be executed in a thread within that JVM. Because the grid computing need real-time communicates between the server and the client, we need add communication module in service. Also, we need task manager module to manage the task. The execution of a task may be monitored via the task monitor module. The execution of the individual tasks is controlled by the scheduler that may

use various scheduling policies to determine the order in which tasks are run. Resources (CPU time, memory, etc) used during program execution can be monitored by the resource monitor module (Figure 5).

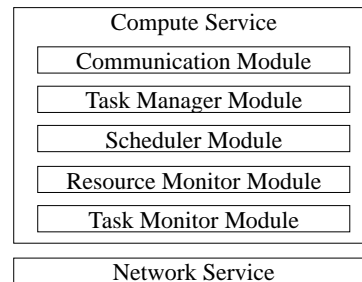


Fig.5 Extend Network Service for Compute Service

Router Service

Because the general Jini system support only to implement distributed computing in a local area network, we need a Router Service to provide and support wide-area service. In this model, the Router Service should like a real Router and it can find other lookup service in wide-area network. Also it should can storage local lookup service information so that other router service can find it. Surely it must have login, query, match module to manage connection request (Figure 6).

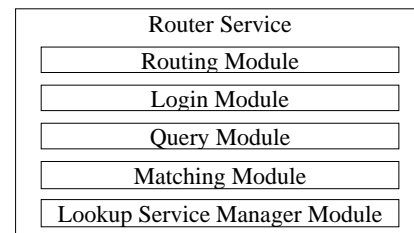


Fig.6 Router Service

Lookup Service

In this model, the Lookup Service is used to simplify the interaction between a client and the grid. It provides advanced search facilities for service location as well as keeps track and caches locally available services. This is illustrated in Figure 7.

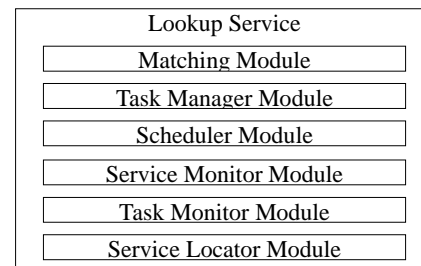


Fig.7 Extended Lookup Service

It discovers Storage and Compute services, stores them in its local service cache. When the client requests services it checks whether the local ones are suitable. If not, the request is forwarded to the grid via the Router Service, and the Lookup Service waits for the search results. Such key tasks are service discovery and program execution. It thus is capable of locating several services that are required for the execution of a complex problem, and then can co-allocate tasks or jobs to these services and control their execution. This functionality is provided by the Matching/Service Locator as well as the task

manager and Scheduler modules.

4. IMPLEMENT MECHANISM

The purpose of the model is to send executable tasks or programs to suitable compute services. So execution always starts by searching for and selecting suitable compute services. The selected Compute Services then receive the executable task or the descriptor of the batch job and start the execution. To begin with, the user starts a client Java program on his local machine. This program connects to the grid via the usual login/discovery mechanism and retrieves a compute proxy object. Using this proxy, the client sends a task along with its resource requirement attributes to the Lookup Service. The lookup service takes the attributes and checks in the Matching module whether there are suitable local services or not. If not, it sends the attributes in a search query form to the Service Locator that will perform a wide-area service discovery operation through the router service broadcasting. Once suitable services are found, the Scheduler will decide which one to use and to which on the task object is to be sent.

Then, the Task Manager is informed to execute the task on the designated service. The Task Manager contacts the Compute service where the local Task Manager receives the execution request. It creates the thread in which to run the object, starts its execution and if required, instructs the task monitor to monitor the execution of the task. At the same time, the resource monitor (monitoring CPU, memory, disk usage) sends information to the registered lookup service so as to update the service state information for the scheduler.

Once the compute service finishes its job, it will send its result to the client through the communication module. Because the Jini system use lease mode, once the service disconnects with the client, the lease will automatically expire. Thus, if the compute service breaks off its thread by chance, the client will send this part of job to another compute service. The complement mechanism is illustrated in Figure 8.

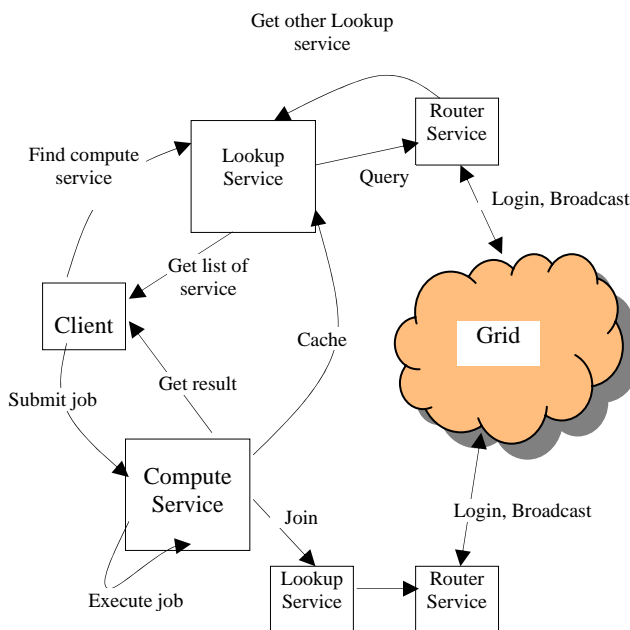


Fig.8 Complement Mechanism

5. CONCLUSION

Jini technology delivers access to services over any network for any platform, any operating system, and any application, regardless of the network complexity, distance, or host device. This means it provides an easy, simple, and fast way to interact with services simply by locating them on a network, with no further action required by the user. Jini technology can be applied to any industry serving commercial, government, community, and consumer markets. Benefited from Jini technology, the paper builds a grid computing model. At present, this model has just a framework. But in principle, this framework should be able to implement and also extend other application. In a word, I believe, Jini technology represents the best hope for consolidating the world brought forth by the revolution spurred by the Web, Java-language-based distributed computing, and consumer electronics. In the end, computers will become what those who worked with them years ago imagined: pervasive, almost invisible, as easy to maintain and expand as your home theater system—in fact, part of it [4].

6. REFERENCES

- [1] W. Keith Edwards, Core Jini (2nd Edition), Prentice Hall PTR, Dec.2000.
- [2] S. Ilango Kumaran, Jini Technology: An Overview, Prentice Hall, Sept.2001.
- [3] JGird project, <http://jgrid.jini.org/>
- [4] Sun Microsystems, Inc. Why Jini Technology Now, Jan. 1999
- [5] Lu Wei, Yang Hui, A New Distributed Computing Platform-Jini and its Application, Journal of Chongqing University (Natural Science Edition), Vol.25, No.3, Mar.2002
- [6] Kuang Zili, Chen Haiyong, Technology of Jini—a Platform of Distributed Computing Based on Java, Journal of Information Engineering University, Vol.4, No.2, Jun.2003



Tang Guosheng, male, born in 1975. He is a master degree candidate of School of Computer Science and Technology, Wuhan University of Technology. His research interests are in Distributed Computing, Databases application technology.

A Layered Distribute Resource Discovery Model in Grid Computing System

Wang Xiaogen¹, Xu Wenbo²

¹Teachers School, Southern Yangtze University

²School of Information Technology, Southern Yangtze University

Wuxi, Jiangsu, China

Email: vctwang@sytu.edu.cn Tel.: 0510-8386802

ABSTRACT

In grid computing system, resources have the characters of dynamic, distributed and heterogeneous. it is important to discover and share resource quickly and effectively. In this paper, we propose a layered distribute resource discovery model of grid system. It takes the advantages of centralized and distribute resource discovery methods. The resource discovery has a high speed; meanwhile the performance requirement to grid device is lower.

Keywords: grid computing, resource discovery, resource management

1. INTRODUCTION

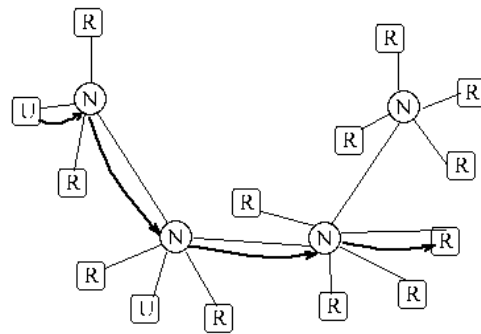
The purpose of grid computing is sharing resource and co-operation in a virtual organization[1]. Grid resources are geographically distributed, heterogeneous and subject to local administration. In a grid computing system, the management of resource is an important function. as the foundation of resource management, how to discover resource effectively and fast becomes very crucial. Resource could not be managed in a good manner if it couldn't be found efficiently.

In a standalone computer, resources could be managed effectively. But in a grid computing system, resources have the characters of dynamic, distributed and heterogeneous, this cause the difficulties in resource discovery[2]. In order to solve these problems, some researchers have given out their solutions, in this paper, we propose a new approach of resource discovery, which is called layered distribute resource discovery model. The paper is organized as follow: First, we discuss the existing resource discovery methods. Then propose our resource discovery method and express it in detail. Finally, we compare our model with the existing resource discovery method, and give out our future work.

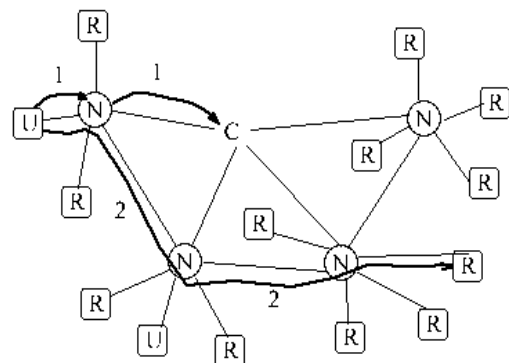
2. RELATED WORKS

There are several approaches of resource discovery in grid computing system. These resources discover method can be classified into two kinds: distributed and centralized. Figure 1 is the principle of these two resource discovery methods. In the figure, R is the resources that can be shared by resource user in grid. We call it resource provider. U is the resource user. N is the connection node between resource provider and user. C is the central node that has the responsibility of resource information storing of the whole grid system. N and C are computers or a network device. Condor's Matchmaker[3] uses centralized architecture to discover grid resource. Resource provider and user register their information to a server. Matchmaker as the central server matches the

information between resource user and provider. It sends matching information to resource provider and user, then the resource user establishes connection with the provider and shares the resource after negotiation. As distributed resource discovery architecture, Globus's MDS-2[4] asks resource provider using registration protocol to register their resource information to GIIS, then a resource user can use query protocol to access resource information from it. Legion uses Collections to locate and discover distributed sharing resources, when resource user query resource information, Legion will search them in multiple Collections. When the resources have been found, Legion will choose one randomly from them and inform the resource provider and user.



Distributed Resource Discovery Architecture



Centralized Resource Discovery Architecture

Fig 1 Principle of Centralized and Distributed Resource Discovery

There are also some another distributed resource discovery approaches. For example, some researchers use P2P architecture to locate and discover the distributed sharing resource[5], some researchers use routing-transferring method to discover sharing resource[6]. All these research works show that distributed resources discovery have advantages than centralized resources discovery in discovery speed. But when the grid computing systems become more and more complexity and extend to wider application areas, using centralized resource discovery methods in a sub grid

computing system will have more merits than a totally distributed resource discovery method. In this paper, we propose a resource discovery method, which not only has the advantages of distributed resource discovery in large area, but also has the merits of centralized discovery method in local sub grid systems.

3. LAYERED DISTRIBUTE RESOURCE DISCOVERY MODEL

3.1 Some definitions

In our resource discovery model, we abstract the grid system into four parts: resources provider, resources user, resource information control nodes and the network. Sharing resources can be divided into dynamic sharing resources and static sharing resources. For example, a super computer or personal computer can be resource providers when they have idle resource. Sharing resources provider or user must be connected to at least one control node. In figure2, A,B,C and D are four sub grid computing systems. Number beside line indicates the distance between two control nodes.

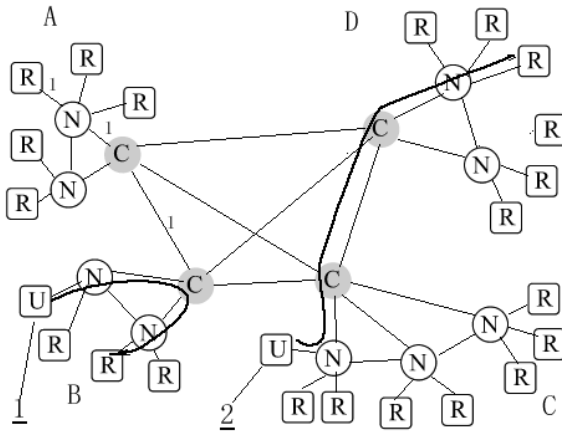


Fig2 Architecture of Resource Discovery Model

3.2 Resource Discovery

When a user looks for sharing resources, it will put forward a request to the central node of local sub grid system, the request includes the information of sharing resource such as resource type and value. The central node will check local resource table first. If the required resource is exist. The central node will notify the user of resource's exact location. The user then will establish connection with resource provider. If the required resource does not exist, the central node will then check the remote resource table, and select a remote central node, which has the required resource information and minimum distance. Then it sends the request to that central node. The remote central node will check local resource table. If existing required resource, it sends the path of resource information back to local central node at once. Local central node tells user the location of required resource. Then the user establishes connection to remote resource provider. If remote central node cannot find required resource either, it will choose one central node around it in the same manner and send the request to that central node, until getting the information of resource provider. The principle of resource discovery is shown in figure 2.

As to local sub grid computing system, it takes centralized

resource discovery method. The central node stores all the resource information of local sub grid computing system, every resource provider must register their resource information to it and update them timely. So the central node can keeps an up to date resource information. Every central node also stores all the resource information of whole grid computing system. The difference between local resource information and remote resource information is the NodeID. The definition of resource information is in table 2. If a resource has the same NodeID with current central node, it means the resource is local. The resource discovery between central nodes takes a distribute method. Every central node stores all resource information of grid system. They take discovery strategy, for example distance by metrics just like RIP, to transfer request each other. In our model, we use the metrics just like the RIP protocol to mark the distance shown in figure 2. When a central node receives a request, it will choose a remote central node from its resource table.

3.3 Recourse table in central nodes

In order to discover a sharing resource correctly, a resource should have a unique identity in the grid computing system. In our model, we use three fields to express a resource item. They are Resource ID, Resource type and Resource value. As the grid computing system is heterogeneous and scalable, integer is used to give the resource a unique resource ID. Resource type includes static and dynamic, so we use Boolean to distinguish them. The resource value field use float value. When the resource is 0(static resource), the resource value will use 0 or 1 to indicate if the resource is existence. Otherwise, a floating value is used to show the resource status. The structure and samples of resource express is shown in table1.

Table 1 Definition and Samples of Resource Structure

Resource ID	Resource Type	Resource Value
Integer	Boolean	Float
1(node)	0 (node)	1 (accessible)
2(memory)	1 (dynamic)	50(MB)
36(software)	0 (static)	1 (accessible)
...

In a central node, it uses a resource information table to store all resource information of grid computing system. A resource information record in this table includes Node ID, Resource ID, Distance, Resource type and Resource value. Node ID indicates the resource's location, which includes the connection node number and the resource number. We use metrics to mark the distance between two central nodes.

Table 2 Resource Information Table in Nodes of Grid System

NodeID	ResourceID	Distance	Resource Type	Resource Value
Integer	Integer	Integer	Boolean	Float
1270	2 (Memory)	5	1 (Dynamic)	50(MB)
350	1 (Node)	8	1(Central)	1(Accessible)
500	36 (software)	4	0 (static)	1(Accessible)

3.4 DISCUSSION

In this section, we will take a comparison between our resource discovery model and other distributed or centralized

discovery model in speed and efficient.

In centralized discovery architecture, all sharing resource information is stored in central node. It has the advantages of lower performance requirements to other nodes except the central node. But the search speed and efficient of sharing resource will go down with the rising of complexity and scale of grid system. Because the resource information stored in central node is probably out of date. When a user gets the information of sharing resource, it needs to confirm the existence of that resource. With distributing resource discovery method, every node in the grid computing system stores all of the resource information. A resource user send request to the node, then the node sends it to its neighbor node. The next node will repeat this operation until reach the resource provider. So the distribute method can find the resource quickly. Because every node in grid system needs storing all resource information, the defect of this architecture is the high performance requirement of node, especially in a large-scale grid computing system. The layered distribute discovery method uses centralized method in local sub grid system and distributing discovery method between sub grid systems. In a local sub grid system, the number of resource provider, resource user and control nodes is limited. So the central node can keep an up-to-date resource information table. When resource user gets the information from central node, it can connect to the provider without confirmation. When central node sends request to another central node, it take the same method liking distributing discovery architecture. The updating speed of resource information in this model is more quickly than distributing architecture, because there are fewer nodes to exchange information.



Wang Xiaogen is a teacher of Teacher's School, Southern Yangtze University. He graduated from Zhejiang University in 1987 with the specialty of Physics. From 1987 to 1992, he worked in Wuxi Machine Tool Equipment Factory as a research member of machine tool's numerical control system project. Then he joined Wuxi MEB Software Engineering Co., Ltd. until 1998. His research interesting includes grid computing, distributed multimedia system, mobile agent, etc.

4. CONCLUSION AND FUTURE WORK

The resource discovery model proposed in this paper has the advantages in higher finding speed, lower performance requirement. It also has good capability in scalable and reliability. We will try to develop a grid-computing model based on it.

5. REFERENCES

- [1]. I. Foster and C. Kesselman, The GRID: blueprint for a new computing infrastructure, Morgan-Kaufmann, 1998.
- [2]. I. Foster, C. Kesselman and S. Tuecke, The anatomy of the grid: enabling scalable virtual organizations, to appear in: Int. J. Supercomputer Applications, 2001.
- [3]. R. Raman, M. Livny, M. Solomon: Matchmaking: Distributed Resource Management for High Throughput Computing. Proc. Of IEEE Intl. Symp. On High Performance Distributed Computing, Chicago, USA (1998).
- [4]. K. Czajkowski, S. Fitzgerald, I. Foster, And C. Kesselman: Grid Information Services for Distributed Resource Sharing. Proc. Of the 10th IEEE Intel Symp. On High Performance Distributed Computing (2001).
- [5]. A. Iamnitchi, I. Foster: On Fully Decentralized Resource Discovery in Grid Environments. International Workshop on Grid Computing, Denver, Co, 2001.
- [6]. Li Wei, Xu Zhiwei, Bo Guanying: An Effective Resource Discovery Method in Grid Environment. Chinese Journal of Computers. 2003 26(11), p1546-1549.

Economic Mechanism Driven Resource Management in Computational Grid

Li Chunlin¹, Lu Zhengding², Li Layuan¹

Department of Computer Science, Wuhan University of Technology, Wuhan 430063, P.R.China¹

Department of Computer Science, Huazhong University Of Science & Technology, Wuhan 430074, P.R.China²

E-Mail: chunlin74@tom.com or jwtu@public.wh.hb.cn

ABSTRACT

In this paper, we apply market mechanism and agent to build grid resource management, where grid resource consumers and providers can buy and sell computing resource based on an underlying economic architecture. All market participants in the grid environment including computing resources and services can be represented as agents. Market participant is registered with a Grid Market Manager. A grid market participant can be a service agent that provides the actual grid service to the other market participants. Grid market participants communicate with each other by communication space that is an implementation of tuple space. In the paper, Grid agent model description is given. Then, the structure of Grid Market is described in details. The design and implementation of agent oriented and market oriented grid resource management are presented in the paper.

Keywords: agent, market, grid, resource management, tuple space

1. INTRODUCTION

Computational Grids are an enabling technology that permits the transparent coupling of geographically dispersed resources (machines, networks, data storage, visualization devices, and scientific instruments) for large-scale distributed applications. Grids provide several important benefits for users and applications: convenient interfaces to remote resources, resource coupling for resource-intensive distributed applications and remote collaboration, and resource sharing. Research in this burgeoning area embodies the confluence of high-performance parallel computing, distributed computing, and Internet computing, attracting successful research from all three disciplines [1]. Several grid systems have been proposed in the last few years. Most of the related systems for Grid resource management such as Legion [13], Condor [15], NetSolve [7], PUNCH [10], and XtremWeb [12] adopt a conventional strategy where a scheduling component decides which jobs are to be executed at which resource based on cost functions driven by system-centric parameters. They do not take resource access cost (price) into consideration. Agent technology is another hot topic in the distributed object-oriented systems [8,9]. Agent can provide a useful abstraction on the grid environment. By their ability to adapt to the prevailing circumstances, agents will provide services that are very dynamic and robust, and suitable for a Grid environment. Agents can be used to extend existing computational infrastructures.

In this paper, we apply market mechanism and agent to build grid resource management, where grid resource consumers and providers can buy and sell computing resource based on an underlying economic architecture. All market participants in the grid environment including computing resources and services can be represented as agents. A grid market participant can be a service agent that provides the actual grid service to

the other market participants. Market participant is registered with a Grid Market Manager. The design goals of our model focus on combining market approach and agent technology to manage computational resource consumers and providers on the grid in distributed style.

The rest of the paper is organized as follows. Section 2 presents the structure of grid market. Section 3 describes market oriented grid resource management. Section 4 gives the related works. Section 5 concludes the paper.

2. THE STRUCTURE OF GRID MARKET

The structure of grid market can be viewed as comprising of a number of nodes capable of running grid service agents, service requestor agents and some infrastructure services. Grid agents interact with each other to achieve a task. To enable Grid Service Agents to be added, updated and removed easily from the grid, some functions are required. This is the role of the "Grid Market Manager" which provides build a virtual market place for resource requestor and resource provider, it also provides communication infrastructure for market participants to communicate with each other. The structure of grid market is shown in Fig.1.

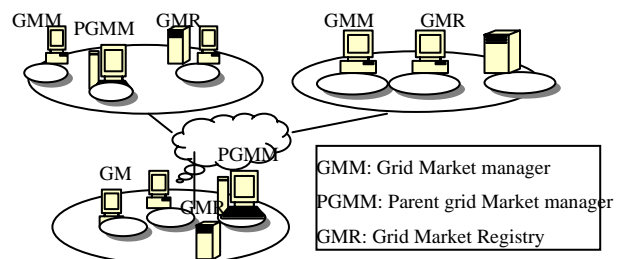


Fig.1 The structure of grid market

2.1 Grid Marketplace

A grid marketplace is a virtual place where one or more resource providers and resource requestors that are called market participants meet to trade on a collection of services. The marketplace is run and regulated by a Market Manager. The Market Manager defines the entry policy to vet participants into the market place, based on participant's identity and/or credentials. The Market Manager is also responsible for putting communication infrastructure in place to ensure that relevant messages are delivered to whom based on participant's identity or credentials. The participants are seeking to strike the best deal for their stakeholders, who are the individuals or organizations that will then have to execute according to the rights and obligations agreed among participants in the marketplace. Participants behave according to a set of rules, which determine what kind of trading takes place. This set of rules is called the market mechanism. The market mechanism of the marketplace is embodied by the negotiation protocol that

participants must comply with during negotiation. The Market Manager is responsible for defining the market mechanism of the marketplace, for enforcing its negotiation protocol, for defining and enforcing admission rules to the marketplace and for providing the communication infrastructure for the marketplace. The role of the Market Manager can be further refined into a set of roles as follows: Market Infrastructure Provider, Provider of the underlying communications infrastructure of the Marketplace.

Grid Market Registry is used to provide a simple registration mechanism common to all nodes and a simple interface for searching the registered services common to all nodes. Grid Market Registry utilizes a "Grid Market Manager" on every node in the grid. This manager is located on a fixed, pre-defined port, which is termed the "Registry Port" and provides a common access point into the Grid Market Registry for all nodes. By locating a manager on all nodes, service requestor agents do not need any localized network information to access the Grid Market Registry. To support the goals of flexible query lookup, implementation of the Grid Market Registry is divided into two components: the Grid Market Registry database and access table. The Grid Market Registry database provides persistent storage and efficient lookup request processing. When coupled with an appropriate database interface, different implementations of the Grid Market Registry can select the most appropriate database based on relevant considerations. A very broad range of persistent Grid Market Registry database interfaces can be implemented. The Grid Market Registry access table is fully resident in memory in the implementation. This access table is rebuilt from data in the Grid Market Registry database as part of a system restart.

2.2 Grid Market Manager

Every node has an individual Grid Market Manager, so the complete system is defined by the interaction between the Grid Market managers. The Grid market Manager allows market participants to discover and utilize the capabilities of other market participants. A market participant may be a service provider, called a *Grid Service Agent*. The Service registers its capability with a Grid Market Manager. A market participant may be a service user, called a *Service Requestor agent*. The Requestor agent discovers Grid Services and requests to use them through a Grid Market Manager. The Grid Market Manager communicates with other Grid Market Managers to perform its role as a service broker. Each Grid Market Manager has a universally unique identifier, *GMM-ID*. GMM-ID is a string used by Requestor agents, Services, and Grid Market Managers to uniquely identify a particular Grid Market Manager in a transport independent way.

Each Grid Service Manager listens on a *service announcement channel*, which is a multicast address set in each Grid Market Manager's announcement. Grid Service Agent's services that want to participate in the system get this address from the Grid Market Manager announcement and then periodically broadcast their own service description to it. Grid Market Manager caches the service descriptions of Service Agent that are advertised. The Grid Market Manager does this by receiving all incoming service announcements using the service announcement protocol that provides service description authentication. The Grid Market Manager adds the description to its database and updates the description's timestamp. Periodically, the Grid Market Manager flushes old service descriptions based on the timestamp of their last announcement.

In addition to receiving service descriptions of Service Agent, another function of Grid Market Manager is to answer requestor agents' queries. A requestor agent uses RMI to connect to the Grid Market Manager, and submits a query in the form of an XML template along with the agent's capabilities (access rights). The Grid Market Manager searches for service descriptions that match the query. Depending upon the type of query, the Grid Market Manager returns either the best match or a list of possible matches.

A Grid Market Manager works together with each other. Each one discovers other remote grid market manager by its Connection Module. For each discovered remote Grid Market Manager, the Connection Module finds the ID code of the remote one, registers the ID code with the local manager, and maintains the association between the transport address and the ID of the remote Grid Market Manager. The Grid Market Manager can discover other remote Grid Market Managers and determine the services registered there. Service Discovery is performed by comparing required services type(s), as specified by the local grid market manager, with the service type(s) available on a remote grid Market manager. Remote Procedure Calls are used to transmit the required service type(s) from the local grid Market Manager to the remote Grid Market Manager and to transmit the response from the remote Grid Market Manager to the local grid Market Manager.

3. MARKET ORIENTED GRID RESOURCE MANAGEMENT

The main actions involved in market oriented grid resource management are grid resource discovery, resource negotiation and resource access. The grid resource discovery provides a basic mechanism to discover grid services for requestor agents. Resource negotiation is responsible for negotiation between grid service agent and service request agent in order to access resource with fees. Resource access is used for service requestor agent to access the located service. This section focuses on grid resource negotiation.

3.1 Resource Negotiation

Negotiation is the process by which service requestor agent and grid service agent interact to reach an agreement through grid market. After an agreement is reached, Service provider and service requestor are given a *Resource Approval*. There are two main roles in negotiation –grid *participant*, and *grid marketplace*. The grid participants are those who wish to reach agreement, and usually they are subdivided into *service requestors* and *service providers*. The grid marketplace described in section 2 is responsible for enforcing the protocol and rules of negotiation. The marketplace is often a third party outside the negotiation. In the case of an auction, the marketplace is the auctioneer. Negotiation consists of the sending of a series of proposals to the negotiation marketplace. Proposals may be sent at any time. If a proposal does not conflict with the negotiation rules, the grid marketplace will accept and process the proposal appropriately. If the proposal does conflict with the rules, the grid marketplace may simply ignore the proposal or it may explicitly reject the proposal. However, in general, the information about the acceptance or rejection of the proposal can be inferred from the information about the negotiation that is made available by the negotiation marketplace to the participants of the negotiation. In general, negotiation consists of the following steps:

- 1) Potential service requestor agents request the grid marketplace for admission to the negotiation. If they are accepted, they receive rules specifying how the negotiation takes place.
 - 2) Service requestor agents submit proposals to grid marketplace according to the rules of the negotiation received at admission.
 - 3) Grid Market Manager validates the proposal against negotiation rules. Every proposal submitted by a participant is checked against previous proposals of the same kind submitted by the same grid participant.
 - 4) If the proposal is valid, the Market Manager forwards then the proposal to the intended addressees. During negotiation, the grid marketplace informs grid participants of the current status of the negotiation, either by sending them current proposals, or by sending some short messages such as the current 'best' proposal (the winner of auction).
 - 5) After negotiation completes, the grid marketplace closes the negotiation locale, and determines any final results.
- The process of resource negotiation is shown in Fig.2.

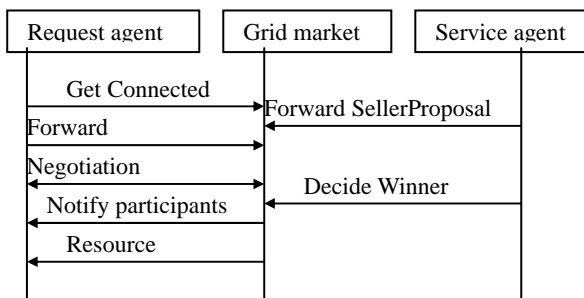


Fig.2 The process of resource negotiation

3.2 Auction Model

Auctions model are proposed as a distributed negotiation mean in our grid resource management systems where both cooperative and self-interested grid agents compete for resources and services. A user endows agent with money for bid, the agent's objective is to complete its jobs as quickly as possible under its budget constrains. To simplify our auction model, we assume that an agent does not have the ability to transfer its money back to its user. We adopt a quasi-linear model for the utility function of each agent. Each agent has a fixed monetary value v for obtaining one or more grid resources. Its utility is the sum of the money it holds and the value obtained from its allocation of resources. Agents wish to maximize their surplus, given the price information provided. Purchasing nothing gives utility 0; purchasing one or more resources at total price x gives utility $v - x$. The agent's money sum v is its maximal ability to pay for the resources, if the price exceeds v , it is better not to buy the resource at all. All agents are risk-neutral; agent never attempts to bid for more than 100% of available resources. Agents have specified starting times, and hard deadlines: all resources are worth nothing after the deadline has passed, so that the utility of a set of purchases of total cost x after the deadline, is $-x$. The auctioneer sets the rules of auction, acceptable for the consumers and the providers. The price of the resources sold by sellers via an auction is not fixed, but it is dynamically determined by the interest of the

bidders. The seller can set a *reserve price*, i.e., a price under which it does not want to sell the resource. Agents can spend time to negotiate the desired resources by using the auction mechanisms, which seem to fit well dynamic and heterogeneous environments. There are several different forms of auction, depending on the number of participants, on the criteria with which the resources are assigned, and so on. Auctions basically use market forces to negotiate a price for the service. The three roles involved in auctions are: resource providers, auctioneers (marketplace), and resources requestor. In grid environment, providers can use an auction protocol for deciding service value/price.

Agents interact with the auctions by submitting bids for resources they wish to buy. A bid is of the form: $((r_1 p_1) \dots (r_n p_n))$. Each pair $(r_i p_i)$ indicates a bid, with r_i indicating resources and p_i indicating the price. When an auction receives a new bid, it sends each of its bidders a price quote specifying the price that would result if the auction ended in the current bid state. Because some offers may be tied at the current price, with some winning and some losing due to tie breaking, the current price is not sufficient for an agent to tell whether it is winning an offer placed at that price. To clarify this ambiguity, the price quote also reports to each bidder the quantity it would buy or sell in the current state. Agents may then choose to revise their bids in response to the notifications (if an agent does not wish to change its bid, then it leaves its previous bid standing in the auction). Communication is reliable but asynchronous; all messages sent eventually reach their recipients. The auction sends the ID of the most recent bid received from the agent with its price quote. An agent responds only to a price quote that reflects its most recent bid sent. Without this method, an agent can have difficulty establishing feasibility, as its understanding of its input and output bid states may be based on none uniformly delayed reports. Bidding continues until quiescence, a state where all messages have been received, no agent chooses to revise its bids, and no auction changes its prices, ask prices, or allocation. At this point, the auctions *clear*; each bidder is notified of the final prices and how many units it transacts in each good.

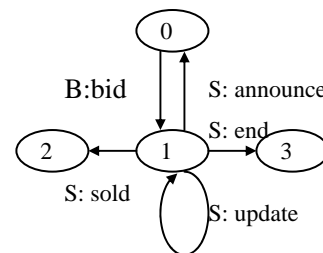


Fig.3 Finite state diagram of auction

The finite state diagram for Auction is illustrated in Fig.3. Auction Protocol is used between a seller agent S and a set of buyer agents B . After the first announce, which is multicast to all agents, if no agent bids (accepts), the auctioneer can end the auction. Otherwise, if an agent decides to try to buy the resource at this price, it sends a bid to the auctioneer. The auctioneer accordingly increases the price, and announces this to all agents. This process repeats, until no agent makes a bid (accept). Then the auction is won by the agent that made the last bid. This agent is told of its win by a sold message and all other agents are informed that the auction is over.

3.3 Grid Resource Discovery

The Grid Service Discovery is used to discover any published resource or service. Resources or services are represented as advertisements. The Grid Market Manager can discover other remote Grid Market Managers and determine the services registered there. Service Discovery is performed by comparing required services type(s), as specified by the local grid market manager, with the service type(s) available on a remote grid market manager. Remote Procedure Calls are used to transmit the required service type(s) from the local grid market Manager to the remote Grid Market Manager and to transmit the response from the remote Grid Market Manager to the local grid Market Manager. Through manipulation of the specification of required service type, the Grid Market Manager can determine characteristics of all the services registered at a remote grid Market manager and the presence of a service on a remote Grid Market Manager matching a specific set of constraints.

Requests of Requestor Agent are expressed in terms of constraints. Constraints represent the requirements for the service to be found. Constraint language of service discovery is based upon XML. Constraints are passed as strings to the constructor of the XML class. During construction, the constraint is parsed and the expression tree is created. The Requestor Agent calls *SearchService()* to ask the local Grid Service Manager to search for Grid Service Managers containing registered services with a particular constraints. The local Grid Market Manager returns the list of GSM-IDs to the Requestor Agent. The Grid Market Manager with the GSM-ID included in the list has information about Grid Service Agent that can provide the service requested by the Requestor Agent. If Grid Market Manager is a parent manager it performs the search on the parent Grid Services Registry that should contain information on all registered services in the grid. If the local Grid Market Manager is a child manager, it will find a local service if a suitable one is available otherwise it will query a parent node for a suitable service. The searching process can also explicitly request a parent Grid Market Manager search. When multiple parent managers are available, the child manager has the option of selecting a parent based on some criteria such as node load. Grid Service Agent provides an XML Description describing its functionality and properties when it registers into the Grid Market Manager. When a service requestor agent wants to use a service, it creates a XML DOM object describing the service it needs along with constraints. Service requestor agent then finds the appropriate service using the XML match. The XML match handles constraints such as requirements, cost etc. For instance, Service requestor agent can ask that the service cost less than some amount, or that it must have some special functions or requirement. Similarly, the matching process ensures that a service requestor agent gets only those services, which it is capable of executing in the required hardware or software environment. Once Grid Service Agent's service has registered with Grid Market Manager, that service is available for use by service requestor agents that will query the Grid Market Manager. To find a service, service requestor agents query Grid Market Manager by invoking the *locate()* method on *Registry* object. The service requestor agent passes a XML service template as an argument to *locate()* method, XML service template serves as search criteria for the query. The *locate()* method sends the XML service template to the Grid Market Manager, which performs the query and sends back many matching service descriptions.

3.4 Grid Resource Access

After negotiation between requestor agent and resource agent completes, requestor agent can start communication with the Grid Market Manager to use the services or resources. Some methods and messages are defined to utilize a service provided by a Grid Market Manager. The local Grid Market Manager is the Grid Market Manager that the requestor agent is calling, and the remote Grid Market Manager is any other Grid Market Manager. The local Grid Market Manager may actually be in a remote node if the Requestor agent is calling the services through Remote Procedure Call.

When a Requestor agent wants to use a service provided by a Grid Service agent under a Grid Market Manager Protocol, it requests a Grid Market Manager to establish a Service Session. The Requestor Agent calls *ConnectService()* to ask the local Grid Market Manager to initiate a Service session with a specific Service agent registered at either the local Grid Market Manager or a remote Grid Market Manager. The Grid Market Manager, with which the specified grid services is registered, calls *ConnectService()* to notify the grid service agent of this *Connect Service* request. The grid service agent may either accept or reject the request. The result is communicated back to the Requestor Agent through the return parameter of the *ConnectService()* call. Once the Service session is established, the Requestor Agent calls *SendMessage()* to ask its respective local Grid Market Manager to send Requestor Agent-specific data to the other end of the Service session. The Grid Market Manager at the other end calls *ReceiveMessage()* to pass the Requestor Agent-specific data received from the other end of the Service session. After Connection session completes, the Requestor Agent calls *EndConnection()* to ask the local Grid Service Manager to terminate the Service session. The specified Grid Market Manager calls *EndConnection()* to notify that the Service Session is terminated. The process of grid resource access is shown in Fig.4.

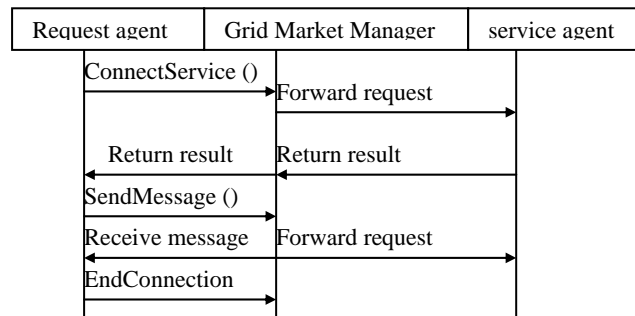


Fig.4 The process of grid resource access

4. RELATED WORKS

Several research systems have explored the use of different economic models for trading resources to manage resources in different application domains: CPU cycles, storage space, database query processing, and distributed computing.

Grid Architecture for Computational Economy (GRACE) [2] proposes a distributed computational economy as an effective metaphor for the management of resources and application scheduling. It proposes an architectural framework that supports resource trading and quality of services based scheduling. It enables the regulation of supply and demand for resources; provides an incentive for resource owners to

participate in the Grid; and motivates the users to trade-off between deadline, budget, and the required level of quality-of-service. It also demonstrates the capability of economic-based systems for peer-to-peer distributed computing by developing users' quality-of-service requirements driven scheduling strategies, algorithms, and systems.

Nimrod-G [4] is a Grid resource broker that allows managing and steering task farming applications on computational Grids. It uses an economic model for resource management and scheduling. Users formulate parameter studies using a declarative parametric modeling language or GUI with the experiment being run on the Grid. Nimrod-G provides resource discovery, resource trading, scheduling, resource staging on Grid nodes, result gathering, and final presentation to the user. Nimrod-G uses GRACE services to dynamically trade with resource owner agents to select appropriate resources. GRACE enabled Nimrod-G has been used for scheduling parameter sweep application jobs on the WWG test bed resources.

The Globus project [14] defines an extensive framework for metacomputing. The project builds a metacomputing toolkit that encapsulates a collection of low-level services (for resource allocation, communication, authentication and file access). Globus is constructed as a layered architecture in which higher-level services can be developed using the lower level core services. Its emphasis is on the hierarchical integration of Grid components and their services.

Compared with above grid projects, our method takes advantage of market oriented model and agent technology to build grid resource management, and provides users with a consistent and transparent interface for accessing grid services. All market participants in the grid environment including computing resources and services can be represented as agents. A grid market participant can be a service agent that provides the actual grid service to the other market participants. Market participant is registered with a Grid Market Manager.

5. CONCLUSIONS

In this paper, we apply market mechanism and agent to build grid resource management, where grid resource consumers and providers can buy and sell computing resource based on an underlying economic architecture. Basic principles of software and knowledge engineering are applied to the development and deployment of agent based grid management. All market participants in the grid environment including computing resources and services can be represented as agents. A grid market participant can be a service agent that provides the actual grid service to the other market participants. Market participant is registered with a Grid Market Manager. Grid market participants communicate with each other by communication space that is an implementation of tuple space. The design goals of our model focus on combining market approach and agent technology to manage computational resource consumers and providers on the grid in distributed style.

6. REFERENCES

- [1] I. Foster and C. Kesselman, *The Grid : Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999.
- [2] R. Buyya, J. Giddy, D. Abramson, *A Case for Economy Grid Architecture for Service-Oriented Grid Computing*, 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), In conjunction with IPDPS 2001, San Francisco, California, USA, April 2001.
- [3] K. Krauter, R. Buyya, and M. Maheswaran, "A Taxonomy and Survey of Grid Resource Management Systems", to appear in *Software: Practice and Experience*, 2001
- [4] R. Buyya, S. Chapin, D. DiNucci, *Architectural Models for Resource Management in the Grid*, First IEEE/ACM International Workshop on Grid Computing Springer Verlag LNCS Series, Germany, Dec. 17, 2000
- [5] Li Chunlin, Lu zhengding, Li layuan. *Design and Implementation of a Distributed Computing Environment Model for Object Oriented Networks Programming*, *Journal of Computer Communications*, Elsevier, Vol 25/5, pp 517-522, Mar 2002
- [6] Nick Antonopoulos, Alex Shafarenko. *An Active Organization System for Customized, Secure Agent Discovery*, *The Journal of Supercomputing*, 20, 5-35, 2001
- [7] H. Casanova and J. Dongarra, *NetSolve: A Network Server for Solving Computational Science Problems*, *International Journal of Supercomputing Applications and High Performance Computing*, Vol. 11, No. 3, pp 212-223, Sage Publications, USA, 1997.
- [8] Li Chunlin, Li Layuan, *Agent Framework to Support Computational Grid*, *Journal of System and Software*, Elsevier, Vol 70/1-2 pp. 177-187, February, 2004
- [9] Li Chunlin, Li Layuan, *Integrate Software Agents And CORBA In Computational Grid*, *Journal of Computer Standards and Interfaces*, Elsevier, Vol 25/4, pp. 357-371, August, 2003.
- [10] R. Buyya, D. Abramson, J. Giddy, *Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid*, *International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, Beijing, China. IEEE Computer Society Press, USA, 2000.
- [11] O. F. Rana and Luc Moreau, *Issues in Building Agent-Based Computational Grids*, *UK Multi-Agent Systems Workshop*, Oxford, December 2000.
- [12] G. Fedak, C. Germain, V. Néri, and F. Cappello, *XtremWeb : A Generic Global Computing System*, *Proceedings of the 1 st IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, May 15-18, 2001, Brisbane, Australia, IEEE CS Press, USA, 2001.
- [13] S. Chapin, J. Karpovich, A. Grimshaw, *The Legion Resource Management System*, *Proceedings of the 5 th Workshop on Job Scheduling Strategies for Parallel Processing*, April 1999.
- [14] I. Foster and C. Kesselman, *Globus: A Metacomputing Infrastructure Toolkit*, *International Journal of Supercomputer Applications*, 11(2): 115-128, 1997.
- [15] M. Litzkow, M. Livny, and M. W. Mutka, *Condor - A Hunter of Idle Workstations*, *Proceedings of the 8th International Conference of Distributed Computing Systems*, June 1988.
- [16] Li Chunlin, Li Layuan,, *Competitive Proportional Resource Allocation Policy For Computational Grid*, *Future Generation Computer Systems*, Elsevier, Vol 20/6 pp. 1041-1054, 2004
- [17] Li Chunlin, Zhengding Lu, Li Layuan, *Apply Market Mechanism to Agent-Based Grid Resource Management*, *International Journal of Software Engineering & Knowledge Engineering*, World Scientific Publishing, Vol. 13/ 3, pp. 327-340, June, 2003

A Taxonomy of Data Location Themes for Storage Cluster and Storage Grid

Yong Feng, Yan-yuan Zhang
Computer Science & Engineering School, Northwestern Polytechnical University
Xi'an, Shaanxi 710072, China
Email: fengyong@co-think.com Tel.: 86-29-88495821-102

ABSTRACT

In storage cluster and storage grid, data location theme is important for improving availability, performance and manageability, but its complexities are poorly understood. There are too many conflict requirements. And possible optimizations are infrequently taken advantage of. To address these difficulties, some assessments criteria will be presented firstly, and then the design spaces are demystified by presenting taxonomy of the themes in use. After analyzing and evaluating a number of popular data locating themes with the taxonomy and assessment criteria, it is expected that data location theme designers can pick one or integrate some of algorithms explained in this paper, even create a new data location theme based on them, to meet their requests.

Keywords: data location themes, storage cluster, storage grid, taxonomy, assessments criteria.

1. INTRODUCTION

Data has become the central asset for companies and organizations, and a dramatic growth of storage capacity and performance request can be observed. Therefore the strategy of managing and storing data is of great importance for companies and organizations. The hardware and software of the storage infrastructure should ensure a fast and safe access to the corporate data. Due to the exponential increase of information, this task can only be accomplished, if the management of the storage infrastructure is centralized and highly automated.

Many kinds of hardware infrastructure, including SAN (Storage Area Network)[1], NASD (Network Attached Storage Device)[2], OSD (Object Storage Device)[3] etc, are involved in connecting dispersed storage resources. They disband the traditional tight coupling of the servers and storage systems, and make a real any-to-any communication among servers and storage systems a reality.

To use the full potential of the interconnected storage resources, it is necessary to integrate storage management solutions into the storage concept. The conventional approach is to tightly couple one or more logical partitions of a disk or raid system with a file system. This strategy has serious drawbacks concerning the efficient use of the storage systems. Furthermore, the administration of this tightly coupled system is error-prone, and it limits the scalability of the storage infrastructure. The efficient use can be significantly enhanced by the integration of a virtualization solution with storage cluster [4] and storage grid [5]. The virtualization solution distributes data among the connected storage subsystems and makes a transformation between an object, which stands for a logical address space presented to the applications, and the access to the physical disks.

Addition or failure of disks and optimization or maintenance of storage system will require a reconfiguration of data layout. Reassigning data to new locations adversely affects the availability and manageability of the system. To accommodate the fact that storage resources and data locations are dynamically changing, this paper studies the current designs of adaptive data placement in storage cluster, distributed file system, peer-to-peer network, and other related domains, and analyses various design choices, then a taxonomy of data location themes is presented to help design a specific data locating theme for storage cluster or storage grid with special requirements.

The rest of the paper is organized as follows: Section 2 describes current storage architectures and assessment criteria for data locating themes. Section 3 presents taxonomy of data locating theme and the related design choices. After analyzing and evaluating a number of popular data locating themes with the taxonomy and assessment criteria in section 4, the conclusion will be drawn in the fifth section.

2. ARCHITECTURE AND ASSESSMENT CRITERIA

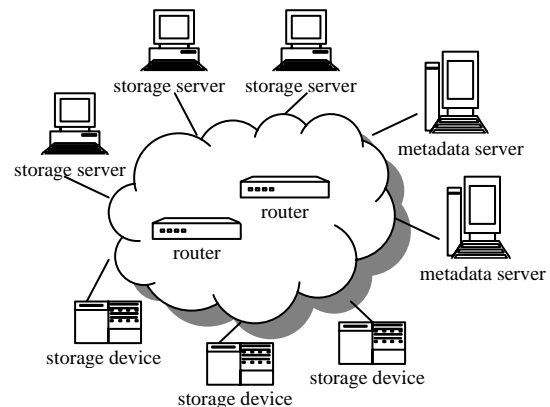


Figure 1: The architecture of storage cluster and storage grid

Storage cluster and storage grid are composed of storage devices, storage servers, metadata servers, and routers, which are attached to a high bandwidth network, as shown in Figure 1. Storage servers provide storage service for applications. Applications store data to and get data from the virtual address space. The forms of virtual address are <object-id, offset> or <volume-id, offset>. Through computing some pre-defined mathematical equations or looking up from a mapping table, the transformation of logical address space to the real devices and physical address can be achieved.

There are always changes in storage cluster overtime. For example, existing devices may exhaust or fail; new devices can join as the demand for storage capacity and performance increases; new logical volumes or objects may be allocated, or

data layout will be adjusted for better performance and utilization. We envision that storage cluster and storage grid have 24x7 availability. Namely, the configuration change in data layout should have little or no impact on the system availability. Furthermore, we also expect the location of data being computed fast with limited space and communication requirements.

The following assessment criteria are considered for data location themes.

Scalability and adaptability

Scalability is the basic requirement for storage system. It means that without disturbing current data accessing a storage system can absorb new adding disks transparent to application, and balance the workload through migrating dataset to new adding disks. In addition, a good location theme is capable of quickly adjusting the data placement to insertions or removals of devices and dataset, if adding or removing devices result only in a small fraction of redistributions and metadata change. For any operation that represents the insertion or removal of a device or a dataset, the number of replacements of data blocks performed by data location scheme can be used to evaluate this kind of ability.

Performance and reliability

Performance is always the overriding concern for the storage system, which supports today's I/O-intensive applications. Data access performance can be further broken down into two parts: data location performance and data transfer performance. We focus on optimizing data location performance for fast lookup and placement in a large-scale storage system. Reliability is also an important aspect to be considered. A centralized server with fast lookup is not acceptable because it is a potential single point of failure. Time complex of algorithm to locating data, and the mount of messages communication when locating data, both can be used to evaluate data location performance.

Balanced utilization

The system should be able to effectively utilize the available storage resources, such as space, bandwidth, throughput, and so forth. When the adding and removing of storage devices or the allocating and deleting of logic storage objects occur, the distribution of data and request in the storage system will become unbalanced, and some storage resources may not be usable due to the restriction of mapping method. Good location theme can support not only space balance but also bandwidth balance and throughput balance.

Management overhead

Store of the metadata for the location information will consume memory of storage system. As the storage system configuration changes, the metadata for the location information may also need to be changed. In addition, to maintain the data locations in a consistent state, data migration is necessary. During this process, the system will consume various resources to maintain the consistency of the state information. It is desirable to keep such maintenance cost low. The mount of the consumed memory and CPU for storing and maintaining metadata is considered as manage overhead.

3. TAXONOMY

When we make a survey of the themes used in practice, we

find a wide variation in assumptions, technique, and the degree to which assessment criteria are achieved. Additionally, many of the design choices are quite intricate and subtle, as we discover when we run the taxonomy past practitioners in the field. Accommodating these conflicting assessment criteria exposes many design options for data locating. They can be classified based on the following points of view.

Smarter versus simple storage device

The storage devices in various storage systems have different intelligence, which determine to what degree the storage device is involved in locating data. Some storage devices are smarter. They have the computation and intercommunication power to support a distributed layout. Some devices, like legacy disks and RAID system, only have ability to attach to the network. According to the ability of storage devices, there are three kinds of data location themes.

For a simple storage device, the storage server locates data for application and sends the request to the right device, sometimes gets help of metadata servers or routers. Since storage device cannot judge whether it owns desired dataset, the resolved address must be precise. Most SAN storage devices belong to this kind of storage device.

For a storage device that can resolve address from a name for dataset it owns, the storage server only need to dispatch the name to storage device. If a dispatch goes wrong, it can be noticed, and the storage server will do it again. OSD is a representative of this kind of storage device.

For a storage device, which can communicate with others, resolving address can be done by itself. Even though the storage server sends requests to a wrong storage device, the requests can be redirected to storage device that owns the desired dataset. The storage grid composed of computers with local disks attached represents this kind of storage device.

Local-scale versus wide-scale storage system

A local-scale storage network has different characteristics from what a wide-scale storage network has, such as elements dimensions, and network performance.

The number of storage devices and data sets determine whether a data location theme is feasible in a way. For instance, in Petal [6], the request should be translated from $\langle \text{vdiskID}, \text{offset} \rangle$ to $\langle \text{serverID}, \text{diskID}, \text{offset} \rangle$. Each node maintains a global map of size $O(N)$, where N is the number of virtual disk in the system. In a Local-scale system, N is typically small and it can afford a global picture of the set of virtual disks. But in a wide-scale system, it is impossible. Algorithms based on mathematical equations are more popular in the wide-scale system, like randomized hashing [7], liner hashing (LH*)[8], and distributed consistent hashing (DCH)[9], for they consume less memory and cost less to keep metadata consistent.

In addition, no global information about resource availability can be provided in large and unstable environments, like storage grid. Every storage node has only mapping information of itself and of its neighbors. In this situation, a router is used to dispatch request to the most possible storage node among its neighbors, like OceanStore [10].

Symmetric versus asymmetric

The function of address resolving can concentrate in some

storage nodes, named asymmetric mode, or evenly distribute among all nodes, which is called symmetric mode.

In asymmetric mode, metadata servers maintain mapping information and provide data location service for storage server. Since every request will pass to a metadata server, like QoS (Quality of Service) guarantees in Stonehenge [11], and load balance and copy-in-time in Sigma [12], some special functions can be easily supported. In some cases, metadata server appears to storage server as storage system and the storage server addresses read and write commands to the metadata server. Then the metadata server transforms the requests and forwards them to the real storage devices. The disadvantage of these cases is that the metadata server has to transfer the bulk data between storage devices and the storage server. If the storage server only sends address-resolving messages to the metadata server and accesses storage devices directly, like PreSto [13], the workload of the metadata server can be greatly reduced, and the performance and scalability of the whole system is improved. But the storage server should be integrated with a driver module, which translates the requests to virtual address into requests to physical storage device. It is complicated when storage servers are heterogeneous.

Metadata servers are prone to be the bottleneck of the whole system and single point of failure in asymmetric mode. Those can be eliminated by symmetric mode. The symmetric mode is categorized into two types, based on whether the mapping information is replicated or partitioned among storage servers. In the case of replicated mapping information, each storage server maintains a globally data locating algorithm. If data locating algorithm uses mapping table, the mapping table could be very large in a large-scale storage cluster, even though Bloom filters [14] can compress it. In order to avoid this problem, the mapping information can be partitioned so that each storage server only manages a portion of the mapping information. In the case of partitioned mapping, we designate a storage server for each data object. The storage server is responsible for resolving the location of its object. A lookup operation first determines the host that is responsible for resolving the location of an object, and then through the host to find the storage device that actually stores the object, just like in Petal [6]. Various DHT schemes for P2P systems, such as Chord [15] and Tapestry [16], can also be regarded as data location schemes where mapping information is partitioned among hosts. In DHT, each node keeps a routing table with a small number of entries, which allows it to forward a lookup request to the host that is closer to the object than itself. An important advantage of the DCH schemes over the traditional partitioned mapping table scheme, is that they can automatically adapt to node additions or departures without any special reconfiguration. One complication in the partitioned approach is that to improve the availability of each mapping table partition, using replication remains necessary.

Controllable versus Uncontrollable data placement

Some schemes assume that physical blocks are placed in a way, which is controllable. The storage system designer or application users can decide data placement policies based on their needs. As a result, the mapping information has to be maintained in a table. If the requirement of controlled placement is relaxed, a function can be used to compute the location of a block, then the mapping table is not needed, which eliminates the need to store mapping information explicitly and avoids consistency issues in maintaining such a

table across multiple nodes. To achieve this computable mapping, DCH [9] or distributed linear hashing LH* [8] can be used. The space requirement is much smaller, ranging from a constant for LH* to $O(H)$ for DCH, where H is the number of storage server. Data lookup only involves computation which varies from a constant for LH* to $O(\log H)$ for DCH. The price we pay for computable mapping is that data placement becomes uncontrollable. Because a hashing function does not consider if there is space available for storing a block or not, a certain amount of free space must always be reserved to keep such a scheme working, which leads to wasted space. Several techniques are proposed to improve storage utilization [7, 8, 13, 17]. However, it remains a hard problem to maintain high storage utilization, especially in a heterogeneous environment where cluster nodes with different capacities are added. In addition, for uncontrollable data placement, when the set of hosts changes, we need to migrate the physical blocks.

4. APPLYING TAXONOMY AND ASSESSMENT CRITERIA

We found it instructive to apply taxonomy and assessment criteria to data location themes described in section 3. In this section, detailed analysis is used to illustrate the design space of data location themes.

GMAP in Sigma

Sigma [12] is cluster file system with global controller. The node of Sigma is a computer with local disks attached. The data object of Sigma is called virtual device, which is an abstract container for a file or a group of files. A 64-bit identifier known as the GID, which is grouped with a locator, identifies each virtual device. The GID and locator are stored in the directory entry that refers to the file. The format of the locator specification is: (GID-(MSPEC)-BLKSIZE-TYPE-REDUNDANCY), where MSPEC is a tuple representing the machines that host the data, BLKSIZE is the block size of the stripe, TYPE identifies the redundancy mode, and REDUNDANCY specifies the level of redundancy. Hierarchical global controller and virtual device controller perform actual striping function for a virtual device. Sigma can control redundancy of a virtual disk in fine grain, and provide services such as performance monitor and snapshot. The memory to store locator information in the whole systems is $O(N)$ where N is the number of virtual disks. After a new node is added, the balance process only proceeds when a new virtual device is created. Rebalancing will change the locator information and make corresponding directory entry of the virtual device locator out of date.

Vdir-Gmap-Pmap in Petal

In Petal [6], computers with local disks are connected by scalable network and provide storage service for application. There are three important data structures: virtual disk directory (Vdir), global map (Gmap) and physical map (Pmap). Vdir and Gmap are global data structures that are replicated and consistently updated on all the servers. Pmap is local to each server. Vdir and Gmap translate the client-supplied virtual disk identifier into server identifier, and then Pmap at the specified server translates the Gmap id and the offset to a physical disk and an offset. Each server of Petal requires memory of $O(N/H)$ to store metadata, where N is the number of virtual disks and H is the number of servers. Locating data requires $O(1)$ communication and $O(\log N)$ times of looking up.

Vdir-Gmap-Pmap cannot automatically adapt to cluster member changed, as GMAP does.

Bloom filter

Being used to determine whether an element can be a part of a set, bloom filter [14] is a compact probabilistic data structure. When bloom filter is applied to a test, it may return true for elements that are not actually in the set, but will never return false for elements that are in the set. In bloom filter, a large bit vector with m bits is used for set E . If E is non-empty, k bits are set for each element in the set. The indices of these k bits are obtained by using k hash functions ($H_k: E \rightarrow \{0, \dots, m-1\}$). Bits can be set multiple times. In order to test if an element e may be a member of E , the k bits $H_k(e)$ need to be checked. Bloom filter can be used in the storage cluster, which uses smarter storage devices and adopts a controllable data placement, to reduce the memory consumed by mapping table [10, 18], but space requirement may still be highly demanded in wide-scale storage system.

DHT in Tapestry

Tapestry [16] is used in wide-scale storage system. The node of tapesry is computer with local disks. It functions as storage server, storage device and router in symmetric mode. The name of data objects and the name of nodes are independent of their location in the form of random fixed-length bit-sequence represented by a common base. Data objects are roughly and evenly distributed in both nodes and object namespaces, which can be achieved by using the output of DHT. Each node uses local routing maps to incrementally route a request to the destination ID digit by digit (eg, $***8 \rightarrow **98 \rightarrow *598 \rightarrow 4598$), which is similar to routing in IP address architecture. This method guarantees that any existing unique node will be found within at most $\log_b N$ logical hop, in a system with an N size namespace using IDs of base b . Every single local routing map assumes that the preceding digits all match the current node's suffix, consequently every node only needs to keep a small constant size b entries at each route level. The local routing map has a fixed constant size of $b \cdot \log_b N$. Because routing only requires nodes to match a certain suffix, tapestry can route around any single link or server failure by choosing another node with a similar suffix. One of the disadvantages of tapestry is the poor adaptability due to its static nature. The other is that the potential troubles spots cannot be corrected before they cause overload or congestion problems over the wide-area. Besides, locating data need high network communication overhead.

LH*

Basically, data objects of LH* [8] are divided into constant-sized buckets. The storage device is a collection of buckets, addressable through a pair of hashing functions h_i and h_{i+1} ($i=0,1,2,\dots$ and i is also called bucket level). The function h_i hashes keys on $N \cdot 2^i$ addresses (N is the initial number of buckets, $N \geq 1$). When existing bucket capacity is exceeded in a storage device, the function h_i of the device is linearly replaced with h_{i+1} , and a new device is absorbed with function h_{i+1} . A special value n is used to determine whether h_i or h_{i+1} should be applied to a key. Storage server computes the device that a bucket is stored in or a bucket should be allocated to. The used algorithm can be found in A1, in which C is the key of bucket, a the device and i the bucket level stored in server.

$$\begin{aligned} a &\leftarrow h_i(C); \\ \text{if } a < n \text{ then } a &\leftarrow h_{i+1}(C); \end{aligned} \quad (A1)$$

The storage device uses algorithm A2 to check whether it should be the actual recipient. In storage device each bucket retains its level, named j . If the result is $a = a^*$, the device is the correct recipient. Otherwise, it forwards the request to device a^* . The device a^* reapplies A2 with its local values of j and a .

$$\begin{aligned} a^* &\leftarrow h_j(C); \\ \text{if } a^* \neq a \text{ then} \\ &\quad a^{**} \leftarrow h_{j-1}(C) \\ &\quad \text{if } a^{**} > a \text{ and } a^{**} < a^* \text{ then } a^* \leftarrow a^{**}; \end{aligned} \quad (A2)$$

LH* only needs space of $O(1)$ to store function and CPU-cost of $O(1)$ to compute address, and there is no master site, through which address computations must go. Accessing and maintaining metadata of LH* are primitive and never require atomic updates to multiple servers. When buckets expand to new device, half of object on old device will be moved to the new device. The mount of data transferred is $O(N/H)$, where N is the number of buckets. Furthermore LH* does not take performance into consideration and cannot rebalance the objects immediately, when the new device is added. The placement of data is also uncontrollable, especially i and n require a centralized coordinator to control the update, which is a potential single-point-of-failure.

DCH

DCH [9] has the similar structure to LH*. It is specialized in solving the problem of different “views” in replication data environment, in which each server may see different set of devices. The “view” is defined to be the set of devices, of which a particular server is aware. DCH constructed hashing functions with following consistency properties. Firstly, when a device is added to or removed from the set of devices, the less amount of the fraction of objects, which should be moved to a new device, is preferred. Secondly, over all the server views, the total number of different devices, to which an object is assigned, is small. Finally, over all the server views, the number of distinct objects assigned to a particular device is small. The space requirement of DCH is $O(H)$, where H is the number of storage server; CPU-cost for computing address is $O(\log H)$; the mount of data transferred for adapting to adding device is similar to LH*. Although LH* seems more efficient than DCH, DCH has no single-point-of-failure.

Randomized hashing in NASD

NASD [7] use thousands of “smart disks” directly attached to the network. Each disk manages its own storage allocation, and allows storage server to store objects, rather than blocks, on the disk. Objects can be any size and may have any 64-bit name. Randomized hashing assumes that each object can be mapped to a key x , which is shared by hundreds or thousands of objects. Objects set is used as allocating unit. It also assumes that disks are added to the system in clusters. Let the j th cluster of disks contains m_j disks. To preserve the balanced load, adding m more disks requires relocating $N \cdot m / (n_j + m)$ objects to the new disks, where a system contains N objects

and n_j disks ($n_j = \sum_{i=0}^{j-1} m_i$). A simple algorithm for mapping

objects to servers without supporting replication or weighted allocation is shown in A3, where existing clusters are numbered $0 \dots c-1$, cluster c is added, and the c th cluster contains m_c disks with n_c disks already in the system.

```

j = c
while (object not mapped)
  seed a random number generator with the object's
  key x. advance the random number generator j steps.
  Generate a random number  $0 \leq z < (n_j + m_j)$ 
  if  $z \geq m_j$  then
     $j \leftarrow j-1$ 
  else
    map the object to server  $n_j + (z \bmod m_j)$  (A3)

```

Randomized hashing has time complexity of $O(nr)$, where n is the number of server addition, and r is the time needed to generate an appropriate random number. Storage server locates a data object without consulting a central server or maintaining a full mapping of objects to disks. This algorithm is probabilistically optimal in both distributing data evenly, and minimizing data movement when new storage is added to the system. This algorithm has space complexity of $O(N/H)$, where N is the number of objects and H is the number of storage devices. It also supports weighted allocation and variable levels of object replication, a little global configuration data is however necessary. Randomized hashing is also used in Storage Tank to manage metadata [17].

5. CONCLUSION

By providing taxonomy, we explore the complications and nuances of data location problems and the design space of solutions in this paper. A number of popular data locating themes are analyzed and evaluated in details. The basic algorithms used in data location themes can be roughly divided into two types: mapping information and mathematical equations. Mathematical equations are better than mapping information, in terms of compactness and stability. Consequently mathematical equations can be efficiently replicated and kept up-to-date across the server. But they can only guarantee the uniform distribution of the hashed values, and mostly rely on the underlying hash functions to provide load balance. Mathematical equations are not sensitive to object workload heterogeneity, and cannot maintain load balance in the situation, where objects have heterogeneous access costs and frequencies.

In sum, no data location theme can achieve all the assessment criteria. The balance of the requirement and cost is thus always the consideration.

REFERENCE

- [1] Tom Clark, Designing Storage Area Networks, New York: Addison-Wesley press, April 2001.
- [2] G. A. Gibson, R. Van. Meter, "Network attached storage architecture", Communications of ACM. Vol.43, No.11, 2000, pp.37-45.
- [3] R. W. Schrock, Object Storage Architecture, Panasas, White Paper, 2003.
- [4] J. Chuang, M. Sirbu, "Distributed network storage service with quality of service guarantees", Journal of Network and Computer Applications, Vol.23, No.3, 2000, pp.163-185.
- [5] Geoff Hayward, Storage Grid: Grid services between storage subsystems on Storage Wide Area, Networks, YottaYotta, White Paper, 2003.
- [6] E. Lee, C. Thekkath, "Petal: Distributed virtual disks", In Proceedings of the 7th International Conference on Architectural Support for Programming Languages and Operating Systems, 1996, pp.84-92.
- [7] R. J. Honicky, Ethan L. Miller, "A fast algorithm for online placement and reorganization of replicated data", In Proceedings of the 17th International Parallel & Distributed Processing Symposium, France, 2003, pp.57-67.
- [8] W. Linwin, D. Schneider et al, "LH*-A scalable distributed data structure", ACM Transactions on Database Systems, Vol.21, No.4, 1996, pp.480-525.
- [9] D. Karger, E. Lehman et al, "Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the World Wide Web", In Proceedings of ACM Symposium on Theory of Computing, 1997, pp.654-663.
- [10] J. Kubiawicz, D. Bindel et al, "OceanStore: An architecture for global-scale persistent storage", In Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, 2000, pp.190-201.
- [11] Lan Huang, "Stonehenge: a High Performance Network Storage Cluster with QoS Guarantees", New York: Stony Brook University, 2003.
- [12] JD Bright, JA Chandy, "A Scalable Architecture for Clustered Network Attached Storage", IEEE Symposium on Mass Storage Systems, 2003, pp.196-206.
- [13] A. Brinkmann, K. Salzwedel et al, "Compact, adaptive placement schemes for non-uniform requirements", IEEE Symposium on Mass Storage Systems, 2003, pp.290-298.
- [14] Burton Bloom, "Space/time trade-offs in hash coding with allowable errors", Communication of the ACM, Vol.13, No.7, 1970, pp.422-426.
- [15] I. Stoica, R. Morris et al, "Chord: A scalable peer-to-peer lookup service for Internet applications", In Proceedings of the 2001 Conference on Applications, Technologies, Architecture, and Protocols for Computer Communications, 2001, pp.149-160.
- [16] B. Y. Zhao, J. Kubiawicz et al, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Tech. Rep. UCB/CSD-01-1141, University of California at Berkeley, 2001.
- [17] C. Wu, R. Burns, "Handling heterogeneity in shared-disk file systems", In Proceedings of the International Conference for High Performance Computing and Communications, 2003, pp.14-26.
- [18] Hong Tang, Tao Yang, "An efficient data location protocol for self-organizing storage clusters", In Proceedings of the International Conference for High Performance Computing and Communications, 2003, pp.1-13.



Feng Yong is a Doctor Candidate of Software Engineering Institute in School of Computer Science and Engineering, Northwestern Polytechnical University. He received his Master Degree from Northwestern Polytechnical University in 2002 with specialty of computer software and theory. He has participated in the storage network management research

project cooperated with storage management group of NEC System Technology Co. since 2000. His research interests are in high performance network storage system and parallel I/O,

fault tolerance, parallel and distributed computing, performance evaluation and benchmark.



Zhang Yanyuan is a Full Professor and a head of Software Engineering Institute in School of Computer Science and Engineering, Northwestern Polytechnical University. He received his Master Degree from Northwestern Polytechnical University in 1987 with specialty of computer software and theory. He is a member of the Committee of Information

Storage Technology, China Computer Federation. He has published four books, over 20 Journal papers. His research interests cover software engineering, storage management, fault tolerance, and distributed parallel processing.

Research on the Data Grid Griddaen Architecture Based on Grid Middleware

Quan Long Quan Liu

School of Information Engineering, Wuhan University of Technology

Wuhan, Hubei, 430070, China

Email: qliu@public.wh.hb.cn longq@mail.whut.edu.cn Tel: 027-87299697

ABSTRACT

Grid computing has become to an available approach to solve the problem about accessing and managing the great capacity of datum. Based on the research of technology and theory of grid middleware, this paper proposes a data grid Griddaen architecture, and describes its total function service and system structure. The data grid Griddaen architecture based on grid middleware can make it enable to integrate the heterogeneous memory resources in WAN environment.

Keywords: Grid, Grid Middleware, Griddaen Architecture

1. INTRODUCTIONS

With the rapid improvement of computer technology, the speed of CPU become quicker and quick, and the ability of processing become more and more powerful in the hardware; while in software, the scope of application program extends constantly, especially the appearance of Internet and WWW, which make the area of computer application wider. On Internet many application programs should run on the heterogeneous platform in the Internet, all of which have made it urgent to develop the new generation software. In this distributed heterogeneous environment, there are many hardware system platforms (such as PC, workstation, minicom and so on), on which there are many different styles of user interface, and also various system software such as different operation system, database, language compiler and so on, however these hardware system platforms even adopt different network protocols and different network architecture connections. Today it is a very realistic and difficult problem that how to combine these systems. The technology of grid middleware is the right solution to this problem, which is the key technology to set up grid system and implement heterogeneous combination.

The middleware architecture makes it enable to compute by grid and combine the existing architectures, it can provide users with coding interface and the corresponding environment, in order to support the development of grid application. The middleware architecture provides the key service, for example process management in the long distance, concurrent distributing resource, memory accessing, information (landing), safety, certification and QoS. Now grid middleware is the focus in the development of the software. Odd-come-shortly, the enterprise software provider Oracle Corporation has provide a middleware product which can predigest the running management of application program in the grid computer environment. This paper proposes a Griddaen middleware architecture that can support data grid. It adopts the distributed multi-domain union server and highly practical technology, so it can support visual files combination and data combination, system data copy, Cache mechanism, can improves the capability of accessing distributed heterogeneous memory system datum.

2. GRID MIDDLEWARE TECHNOLOGY

Normally middleware means an independence system software or service program that runs on the client or server system, and is a new mode to develop software. In the actual application, it can implement many functions, such as provide the course management for long distance, distribute space information resource, store and access information, safely load system and certificate, monitor system safety or service quality and so on. Middleware is understood as a kind of software, not a sort of software. In the grid environment, middleware can implement not only the interlinkage of all kinds of application program easily, but also all sorts of more complex cooperation. Presently based on the all kinds of applications in the distributed environment, we introduce the middleware mainly in order to solve the communication problems on Internet. Thereinto the middleware is located between application layer and network layer. Because the middleware implements functions in each layer and transparently encapsules, it is feasible that the application layer software is independent of the lower layer implementation mechanism (such as computer hardware and operation system platform), we only should develop it separately, then can operate access the different platforms. In the actual existing applications, a large majority of big enterprises all use the middleware technology to build distributed application standard platforms. By using kinds of middleware, it is enable to combine the existing dispersed subsystems in corporations and improve the simplicity and robustness of the system combination^[1].

The basic concept of the middleware is shown in Fig. 1.

If the cooperation among various kinds of application programs in different layers is understood as a client/server mode, the introduction of middleware extends this traditional client/server structure, and forms a new three-layer or multi-layer structure that includes client, middleware, and the server. This structure supports the development of dependable, extendable, complex compressional application^[2].

Generally speaking, from the point of the view of the client/server model, the basic work principle of middleware is as shown in Fig. 2. The application program on client need get some datum or services from some node in the grid architecture, and these datum and services perhaps are from the server that is running with a different operation system from the client's. However in this case, all kinds of application programs only have to access the middleware system, it can automatically finishes the tasks to search the object datum or services, set a client request and recombine the result as an answer information and set back to application programs.

In the environment of distributed grid, the middleware is divided into 4 kinds^[3].

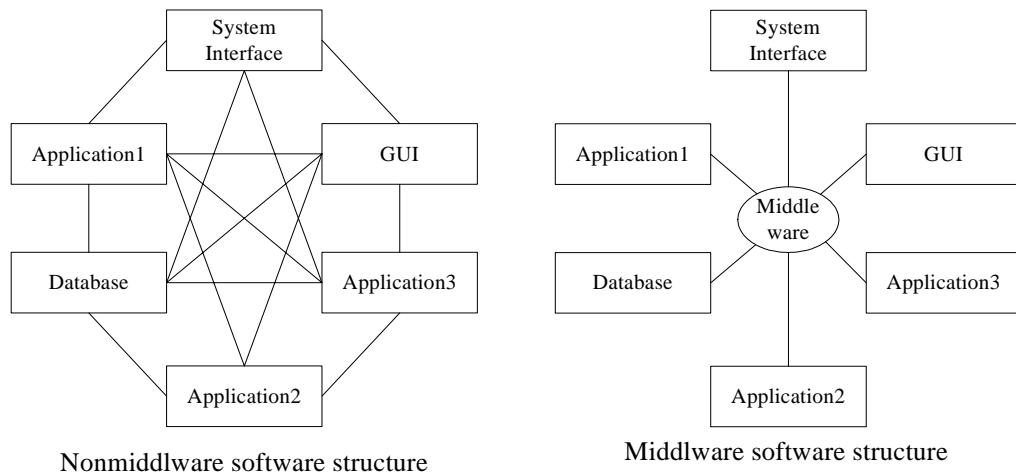


Fig. 1. Basic concept of the middleware

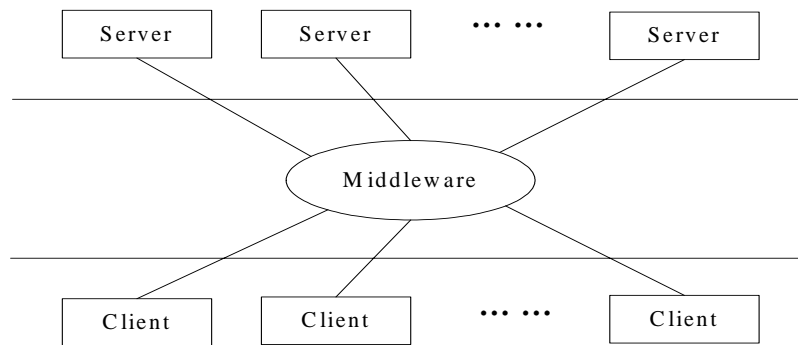


Fig. 2. Basic theory of middleware

- (1) The middleware based on Remote Procedure Calls, is an extension of process transferring of program language, the object that is transferred can be kept on any distributed system physical platform.
- (2) The middleware, faced information, supports the course communication mode that is based on information transfer. This kind of middleware suit not only the client/server mode, but also the point-to-point mode, generally it is more efficient than the middleware based on the RPC.
- (3) The middleware, based on Object Request Brokers, is the preferment for object oriented application program. The information can choose the router by ORB, on the same time ORB solves the problems about the combination and security.
- (4) Database middleware, can pellucidly access to the heterogeneous traditional database. The Griddaen architecture proposed in this paper is a kind of database middleware in Gridippen grid system. It supports the system part in the database grid function, combines all kinds of data file memory system, and provides a seamless accessing mode for distributed datum.

3. THE GRIDDAEN DATA GRID ARCHITECTURE BASED ON GRID MIDDLEWARE

3.1 The Whole Function Service

The Griddaen data grid architecture can combine all the heterogeneous memory resource in WAN environment, for example Linux, Windows; the stand-alone file system, NFS; the network file system and database system and so on, and organize them together. Providing the data accessing and manage service, the system can shield the heterogeneity of bottom resource and too many control fields, which provides the users with a simple integrative file view, convenient normative accessing and operation. The logic function of system service is shown in Fig.3.

Griddaen data grid as a system middleware has three layers: The first layer, faced to material memory resource accessing interface, can access and use the bottom data memory systems that include bottom data memory resource, meta-information resource, all kinds of file system and database systems. It adopts the driver protocol and accessing means that are supported by all the memory systems to access and use datum in the systems.

The second layer is the key service layer. It handles many data sources and unified control the access in the data grid architecture. The second layer includes resource aggregation, data service, meta-data service, security service and system

operation to be quickly transferred, and of copy operation, copy management and virtual data management.

- (1) Resource aggregation, mainly faced to the management with the access, scheduler and service to the resource of computation, device and so on, supports the functions needed by computation grid.
- (2) Data service module mainly provides the services that to optimize the access, scheduler, and deals with the datum on the distributed heterogeneous memory resource into integrative datum. As well as it provides the service to uniformly access datum, allows for the datum the
- (3) Meta-information service provides the systems with the services that includes information service in whole field resources, locating datum and searching properties of datum, registering and issuing datum, querying and vindicating the system resource, handling and choosing copy information and providing the users and system with an interface and protocol for meta-information.

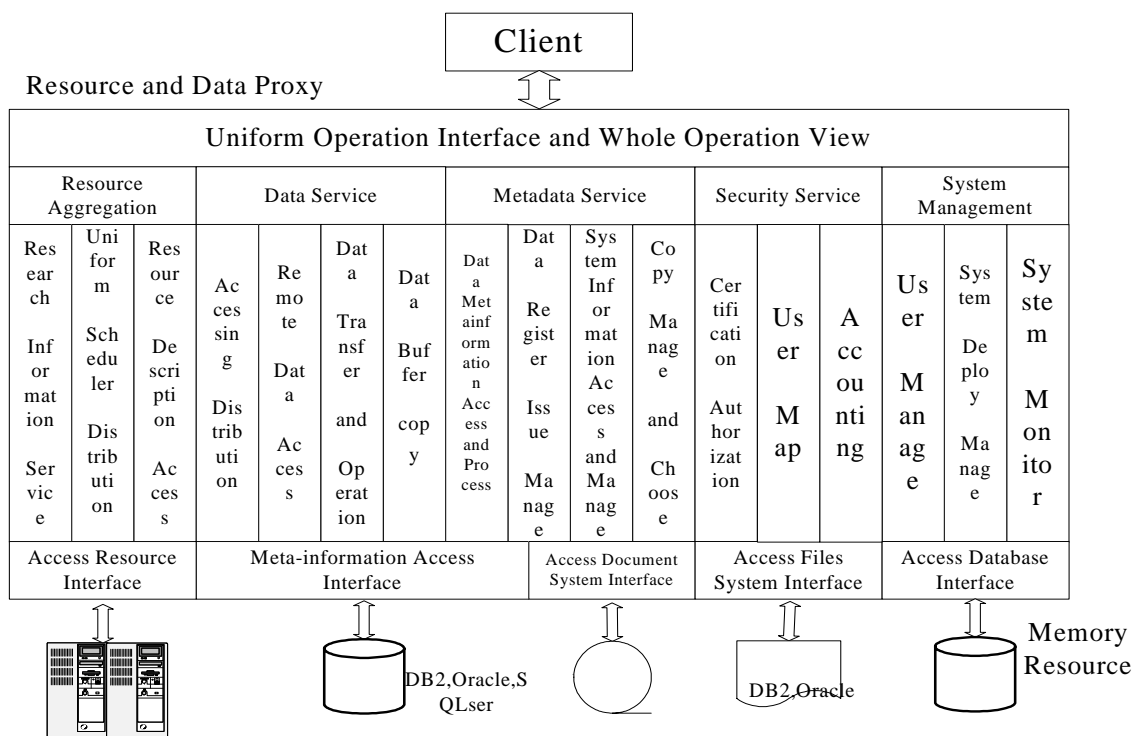


Fig.3 The logic function of system service

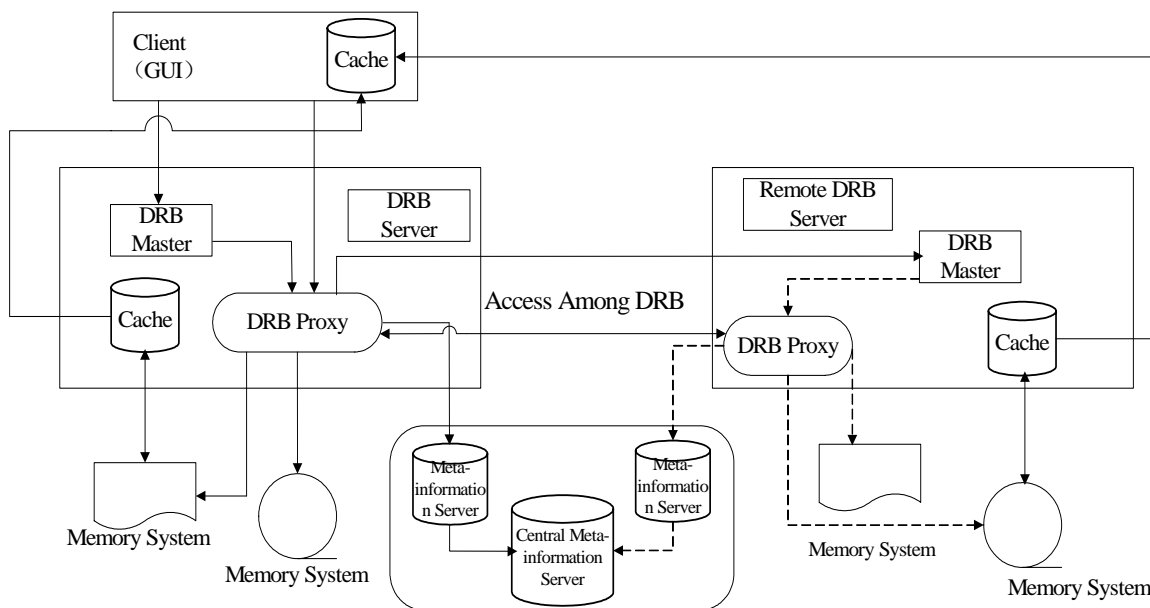


Fig.4. The Griddaen architecture

- (4) Security service mainly supports the single landing certification and multi-layer access control and authorization mechanism.
- (5) System management mainly realizes the operation to found and erase grid system user, collocate and deploy the systems and monitor the state of the whole grid architecture.

The third layer provides the data service that the data grid provides the users with the interface, which makes all the different user interfaces compatible.

3.2 Griddaen Middleware Architecture

The Griddaen architecture is mainly composed of DRB service, meta-data server and so on, as shown in Fig.4. DRB (data Request Broker) provides the user with the functions of accessing, memory and controlling datum. It is a distributed structure, each SITE manage field has an independent DRB server, in order to provide the service that access datum. When the user set a request, DRB Master will produce a DRB Proxy that can provide the users with all kinds of data operation services. The distributed DRBs can cooperate with each other to provide the data service. DRP analyses the request set by the users and transfers this application request to the right memory resource, then starts the DRB in corresponding memory that begins to provide the data operation and management service. After the DRB gets the data, it will transfer the data to the clients by highly speed data transport protocols.

Metadata Server (MDS) is a mutil-layer distributed structure, composed by part meta-information server and central whole meta-information server. Each of the part metadata servers takes charge of the service for local resource and data meta-information, and provides the access to the meta-information service. Central servers build the index of each part meta-information and data buffer, and support the realization to unified access interface and view the whole datum. MDS and DRB are independently designed from each others; build the service relationship through system deploying.

4. CONCLUSION

Grid is the third technology tidal wave after the Internet and the Web, which has broken the information isolated island, and realizes the multi integrative service. It is the grid middleware that is the key technology in the grid. The grid middleware not only makes the development of application system's simple and convenient, shorts the development cycle, but also reduces the maintenance of system and workload of running and handling; in addition it decreases the investment for the computers. The Griddaen architecture proposed in this paper is the architecture based on the middleware technology and can support the data grid. It combines all kinds of data files memory systems, provides a uniform seamless access manner for distributed data. So it is realized to combine all kinds of heterogeneous memory resource in the WAN, and improves the sharing and interoperability of different resources on the Internet.

5. REFERENCES

- [1] Ian Foster, Carl Kesselman, Steven Tuecke, The Anatomy of the Grid, Intl J. Supercomputer Applications, 2001.
- [2] Nik Looker, Jie Xu, Assessing the Dependability of OGSA Middleware by Fault Injection, 22nd International Symposium on Reliable Distributed Systems, 2003.
- [3] Junseok, Hwang, Praveen Aravamudham, Elizabeth Liddy, IRTL (Information Resource Transaction Layer) Middleware Design for P2P and Open GRID Services, Proceeding of the 36th Hawaii International Conference on System Sciences 2002 IEEE.

Long Quan, female, postgraduate student, Wuhan University of Technology. Her research interests are information security and grid computing.

Liu Quan, female, Full Professor, tutor of doctor, dean of the School of Information Engineering, Wuhan University of Technology. She graduated from Huazhong University of Science and Technology in 1985 with specialty of Wireless Communication. She has published three books, over 50 Journal papers. Her research interests are in information security, signal processing, communication technology, grid computing and network security.

The Application of Simulation in Large Scale Traffic Flows System Based on Grid Computing

Yun Wu

College of Management, Huazhong University of Science and Technology
Wuhan, Hubei, 430077, China

Email: wuyun1974@hotmail.com Tel.: 027-8725702

ABSTRACT

In this paper, we study large-scale traffic flows system simulation problem based on computer parallel process technology and object oriented model theory on distributed network. We give a grid computing method to build traffic system simulation model, and apply the visual reality technology of simulation for display. In practice, we use Globus Toolkit (grid computing software tool) and World Toolkit (visual reality software tool) to create a traffic flows system model and to simulate the traffic system. The traffic flows simulation system gives a way for large-scale traffic simulation system on grid computing. We give a preliminary examination and theory guidance about complex large-scale traffic simulation system in distributed environment.

Keywords: grid computing, traffic flows system, simulation, traffic system model, globus toolkit virtual reality

1. INTRODUCTION

When we study the large-scale traffic flows simulation and traffic system plan, the theory, the method and technology on the multi-subject knowledge are needed. Especially the system engineering methods and computer technology are the most important ones, because the traffic system problems include many subsystems, such as, road network system, social activity system etc. When the traffic system plan based on traffic flows system simulation is studied, large amount of data are needed to deal with. As the traffic flows system always has a large and complex scale, we need more advanced data-processing technology, math calculation methods and computer technology. In order to simulate a traffic flows system in a city or more complex traffic system in computer, the parallel process technology and distributed process technology are necessary. There have been some researches in these fields or related fields. In the field of traffic network system theory model, Ran B. (1993)[1] studied dynamic transportation network model for advanced traveler information systems, Yang H. and Bell M. G.(1998) [2]give a review about models and algorithm for road network design. In the field of grid computing, K. Czajkowski, S. Fitzgerald et al (2001) [3] posed a problem about grid information service for distributed resource sharing. W. Allcock, A. Chervenak et al (2001) [4] studied the grid architecture for the distributed management and analysis of large scientific datasets. In the field of traffic flows system simulation Joachim K et al (2000) [5] develop a modularized approach for comprehensive air traffic system simulation. Lino Figueiredo, J. A. Tenreiro Machado et al (2003) [6] studied simulation and dynamical analysis of freeway traffic. In this paper, in section 2, we have discussed the state of the models in traffic flow system theory and given some traffic subsystem models we needed. In Section

3, we have introduced the grid computing methods on the traffic network system simulation. In Section 4, we have studied the display problems about the visual reality software tools, and will give a traffic network system simulation application. The paper is ended with some suggestion.

2. THE MODEL THEORY IN TRAFFIC SYSTEM

In traffic system engineering, more investigated data are needed, especially O-D pair investigated data and social activity data. Based on such data, we can forecast the traffic flows in reality. So we choose some math models on individuals, families, subarea etc, such as, population forecast model, family construct model, ground utility model etc. These models will deal with the original data of traffic flows system and give a reasonable analysis and statistics to find useful reasonable data for traffic system simulation. More studies have been done there. So we choose the models from those researchers and apply it in our software application. We just introduce some most important ones of them briefly.

Population forecast model

$$P'x(1) = \sum_{i=15}^{45} Pf(i)Bx(i) \quad \text{Eq. (1)}$$

$P^1 x$, Px represents the amount of a class of population on the former year and next year. Subscript X can choose m or f, identifying man or woman.
 $Bx(i)$ represents birth rate of x class of people

Family income model

$$\Pr(a < X < b) = \int_a^b \frac{\lambda^{s+1}}{\Gamma(s+1)} x^s \exp(-\lambda x) dx \quad \text{Eq. (2)}$$

Car owned rate model

$$f(x) = \frac{1}{1 + be^{-cx}} \quad \text{Eq. (3)}$$

Family construction model

$$\Pr(n) = \frac{N!}{n!(N-n)!} q^n (1-q)^{N-n} \quad \text{Eq. (4)}$$

City ground utility model

We choose Hansen model:

$$A_i = \sum \frac{W_j}{\exp(rt_{ij})} \quad \text{Eq. (5)}$$

$$N_i = N \frac{A_i^a S_i}{\sum_j A_j^a S_j} \quad \text{Eq. (6)}$$

These models play a key role in pre-processing original data in traffic research. We use them in our software programs to forecast real traffic flows on scientific ground. This method is called regression analytical method.

In the aspect of Patrol allocation forecast, we use gravitation model.

$$q_{ij} = K \frac{P_i^a A_j^\beta}{R_{ij}^\gamma} \quad \text{Eq. (7)}$$

In the aspect of traffic manner partition .we use LOGIT mode

$$P_j = \frac{\exp(\sum_{k=1}^K \theta_k X_{jk})}{\sum_i \exp(\sum_{k=1}^K \theta_k X_{ik})} \quad \text{Eq. (8)}$$

In the aspect of traffic impedance, we use the traffic model of U.S.A road bureau. It is the BPR function.

$$t_a(q_a) = t_a(o) [1 + \alpha (\frac{q_a}{e_a})^\beta] \quad \text{Eq. (9)}$$

And we use beckmenn equilibrium allocation model for traffic flows allocation, etc...

All these models give us theory guidance and math formulas on traffic flows system simulation.

3. GRID COMPUTING APPLICATION ON TRAFFIC SYSTEM

a) Pre-process in Distributed Environment

First of all, we have to solve a partition problem about complex traffic flows system simulation, which is how to divide such complex system to be adapted to distributed environment processing mechanism in the computer networks. According to the feature of traffic system, we can divide it according to traffic plan steps. Since there are four steps, we divided it into four subsystems or parts. We apply each subsystem to different computer in the distributed computer networks, one computer or some common kinds of computers for the task that deal with traffic occurrence simulation, the next for traffic allocation simulation, the third one for traffic manner partition simulation, and the last for traffic plan simulation. The traffic occurrence simulation computer deals with the patrol data gathered by the investigators all over the studied areas. Its preprocessing results will be sent to the traffic allocation simulation

computer, where the traffic allocation simulation computer will analyze the high rank O-D pair matrix. Then the traffic manner partition computer will do its jobs on the basis of the result data sent from the traffic allocation simulation computer and send its result to the traffic plan simulation computer. At last, the traffic plan simulation computer will deal with the reasonable traffic allocation problem. The feedback will react to the former computers in order to get an optimal result.

b) Traffic Simulation System on Grid Computing

The amount of data for traffic system simulation is large. We need to apply different data blocks to different computers. If the coupling degree is small, we can apply the computer parallel processing ability to change it into a large one. For example, we simulate a complex traffic flows system composed by ten thousand traffic units. In theory, each traffic unit has relationship with the other. But in the same area, such as in Wuhan city, the coupling degree of the traffic units belonging to this area will be greater than that of the ones belonging to different areas, such as the traffic units in Beijing city and in Wuhan city. On the basis of this phenomenon, we can use different parallel computers to deal with or simulate the traffic information in different areas. So grid computing can be used in complex traffic simulation system. It is needed to know the traffic information in each contiguous area. In one aspect, it sends traffic jam-up in its own area to the adjacent one. On the other hand, it will receive traffic information changes from other adjacent area. The simulation system uses protocol data unit (PDU) to send data on the networks, because the communication information among the areas is partial. For example in Wuhan city, the traffic unit from Hankou district is only needed to know the jam-up information of Wuchang district and to decide the routine way, and is not needed to know the jam-up information of Beijing city. We can divide the blocks of core model units composed by the adjacent areas. In each core model unit, the traffic unit has strong relationship with others. The core models are feebly related. We put some core models in some computers and put other models in others to run. This will enlarge traffic system scale and avoid exponential increase of communication information and toleration of simulation system.

We use Globus Toolkit to realize the grid computing task on this traffic simulation system. On the complex traffic simulation system, we will install GRAM (Globus Resource Allocation Manager) on each parallel computer. GRAM is responsible for object resource management in local traffic areas and sends the changed information of the local object resource to MDS (metacomputing directory service). In this way, the latest information will be achieved by MDS. Due to cooperation allocation of area traffic object resource, Globus use DUROC (dynamically-updated request online coallocator) to deal with. DUROC is responsible for communication within all the GRAM. After parallel computer gains enough computer resource, it can use GEM (GLOBUS EXECUTION MANAGEMENT) to run executable software program code and traffic data from preprocessing by grid computing. The output data and log from simulation program will be saved into a special computer for display with the help of GASS (global access to secondary storage). Then we can monitor traffic status and display real time traffic status on computer screen by the WTK software tool. The figures will be shown.

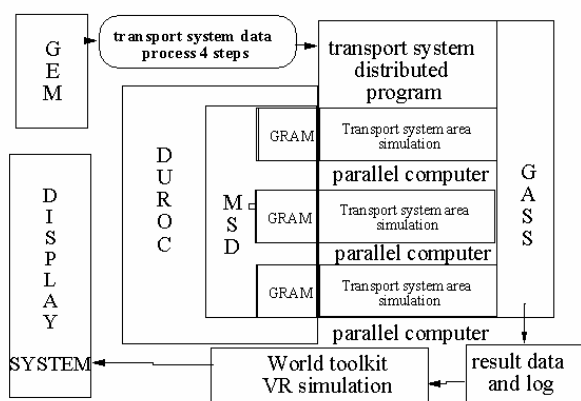


Figure 0

The core program pseudo codes are shown as below.

```

read input data
read input parameters
initialize variables
do I = 1, max_timesteps
solve equations
call globus function to get remote parameters
based on these return code,
update variables with new parameter values
and continue computing
or, re-initialize all variables and restart main loop
or, stop computing and exit
end do
write output data
end program

```

4. DISPLAY BY VISUAL REALITY TECHNOLOGY

We use visual reality technology to display the traffic scene of complex traffic simulation system by C plus plus program language and world toolkit software develop kit. WorldToolKit (WTK) is an advanced cross-platform developing environment for high performance, such as real-time 3D graphics applications. With WTK's function library and end-user program tools, we can develop traffic simulation system applications. We use C plus plus program language to call a library of WTK functions written to develop interactive and real time traffic simulation systems. And WorldToolKit can also support the simulation environment on distributed network, so we use WTK function to simulation our system together with GLOBUS tools, especial the display subsystem. When users change their viewpoints, the simulation system will use WTK matrix transform function to change current position and direction of traffic scenes display. The traffic simulation system will tell the users the latest position and direction and find whether the traffic unit information comes from remote computers, especially the remote traffic unit crossing the local areas. If this kind of information comes, traffic simulation system will add this remote traffic unit to their local model library and obtain traffic information to display traffic scene. Similarly, when the local traffic unit leaves local area, local computer will remove the local traffic unit from local traffic model library and send special traffic information to other areas. This cycle of sending and receiving traffic-specified messages is repeated,

resulting with a distributed simulation. We use special display system center to display real time traffic scene.

The figures are shown as below. In figure 1, the red point represents the jam-up point. In figure 2, the red line represents a heavy-traffic road and the green line represents the free-traffic road.

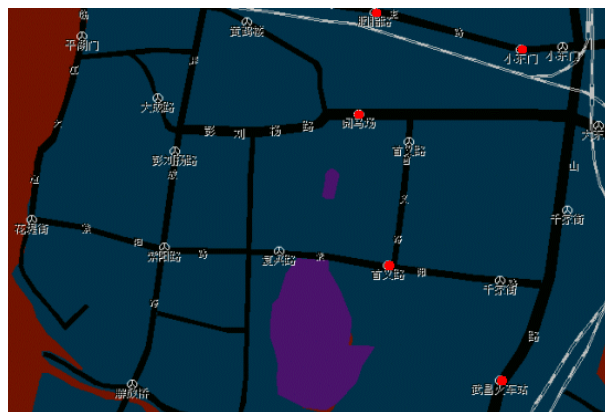


Figure 1

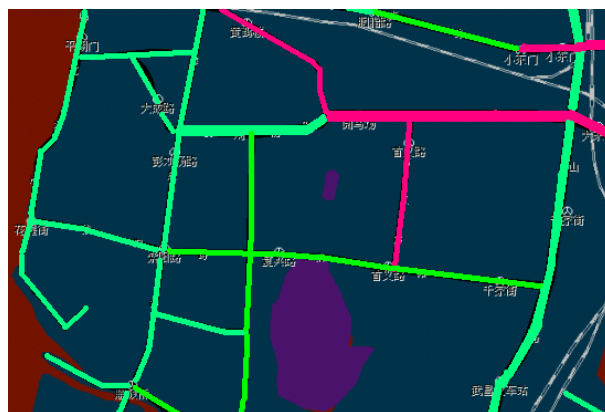


Figure 2

5. CONCLUDING REMARKS

In this paper, we give a preliminary examination and theory guidance on complex large-scale traffic simulation system in distributed environment. We design a traffic simulation system to display the real time traffic scene. But the traffic model and theory are not mature enough, and more studies need to be done in these fields.

6. REFERENCES

- [1] Ran B. (1993), Dynamic transportation network models for advanced traveler information systems, PhD thesis, University of Illinois, Chicago, IL.
- [2] Yang H. and Bell M. G. (1998), Models and algorithm for road network design: a review and some new development, Transport Reviews 18(3):257-278.
- [3] K. Czajkowski, S. Fitzgerald, I. Foster, C. Kesselman (2001) .Grid Information Services for Distributed Resource Sharing. Proceedings of the Tenth IEEE International Symposium on High-Performance

- Distributed Computing (HPDC-10), IEEE Press, August 2001.
- [4] W. Allcock, A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, S. Tuecke.(2001) The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications*, 23:187-200, 2001.
 - [5] Joachim K. Plaettner-Hochwarth¹ & Yiyuan J. Zhao² (2000), a modularized approach for comprehensive air traffic system. Research paper, NASA Ames Research Center August 2000
 - [6] Lino Figueiredo, J. A. Tenreiro Machado (2003): *Simulation and Dynamical Analysis of Freeway Traffic*, PhD thesis, University of Porto, July, 2003.
 - [7] Liuchanqi (2001); *advanced traffic planning*, Beijing, people traffic press, Dec, 2001(in Chinese).
 - [8] Duzhihui, Chenyu, Liupeng (2002): *grid computing*: Tsinghua University press. 2002 (in Chinese).
 - [9] *WorldToolKit develop manual*, sense 8 company, 2001



Mr **Wu Yun** is a Ph.D student in the College of Management, Huazhong University of Science and Technology majoring in Management Science and engineering. He is also a teacher in the College of Management, Wuhan University of Science and Technology.. He has published 7 Journal papers. His

research interests are in MIS, ERP, data warehouse, data mining, visual reality, grid computing, network security and e-commerce.

A Distributed Peer-to-Peer Platform for Synchronized Group Collaboration and Knowledge Sharing

Jo-Yew Tham, *Member, IEEE*; Seng-Luan Lee; Choon-Ee Tan, Roger; and Leong-Chiang Tee

Centre for Industrial Mathematics*, Department of Mathematics

National University of Singapore

2 Science Drive 2, Singapore 117543

Email: thamjy@ieee.org; {matleesl; scitance; mattlc}@nus.edu.sg Tel.: +65 6874 3183

ABSTRACT

This paper presents new research and development of an integrated peer-to-peer (P2P) platform comprising of a network of distributed and decentralized peer devices connected directly with one another in an ad-hoc virtual group manner. The platform is built upon an extended version of the Sun Microsystem's Project JXTA and the Jabber's XMPP (Extensible Messaging and Presence Protocol) protocols. The proposed P2P platform has a comprehensive set of application programming interfaces (APIs) that provide a high-level encapsulation of many core P2P platform services, security control and policies, online presence management, data management and transfer, etc. to the application layer. By using these platform APIs, we have developed a number of essential tools targeting distributed e-Education and e-Collaboration environments, such as integrated secure chat, digital asset management, sharing, searching and retrieval, synchronized calendaring and contacts management, and scalable multimedia communications. The core vision and strategy here is to enable a truly distributed means for multiparty communications and collaboration in an ad-hoc peer group eco-system without the need for centralized systems, file servers, databases and corporate networks setup (such as extranets and virtual private networks). The platform's flexible plug-in and XML web services architecture allows easy development and integration of many new applications and services modules. By employing a 100% Java implementation, the platform is OS-independent and has been shown to work well on Windows, Macintosh, and Linux.

Keywords: Distributed peer-to-peer infrastructure, distributed database, multimedia, and scalable compression, presence management, synchronized group e-Collaboration, Project JXTA, XML Web Services.

1. INTRODUCTION

Consider a world of directly connected networks whereby each online user can actively and conveniently engage in a unified interactive environment to securely exchange his or her knowledge as well as to share, search and retrieve digital assets directly with one another. Such interactions can happen at any time and place using any networked devices, regardless of whether the user is on the public Internet, behind some corporate firewalls, or on mobile networks. In addition, this ad-hoc environment offers state-of-the-art cryptographic security and grants authorized user the total freedom to create ad-hoc protected collaborative groups *without* the regulated

central control and setup of secured extranets or virtual private networks. Hence, this results in near-zero administrative and operational costs due to the absence of expensive centralized servers, databases, and file storage systems, while fully leveraging existing computing hardware resources as the peer devices in this P2P eco-system.

Although there exists a number of client-server solutions that attempt to create these collaborative environments, such a centralized computing solution does not truly add value nor fit into the natural workflow lifestyle demanded by many users. Consider the scenario of knowledge collaboration (i.e., organize, classify, share, search, retrieve, co-edit, post, survey, group chat, etc.) across different corporations or among multiple geographically dispersed offices of a large multi-national company. It is important to note that both the digital assets (data, documents, graphics, audio and video files, presentations, spreadsheets, etc.) as well as the knowledge base of the professionals are naturally originating and residing at their respective computing systems and not automatically pooled together at any centralized servers. By forcing the users to upload all digital assets from the local systems to a centralized repository, just to enable searching and sharing, clearly does not fit into a natural and productive workflow process. More importantly, the knowledge know-how and skill-sets of a professional are intangible assets that cannot simply be stored in centralized locations for later searching and retrieval.

In today's many client-server solutions, it has become habitual that almost nobody shares anything by explicitly uploading his or her assets from the local computers to a central server. At the very most, some assets may be shared but they are oftentimes not the latest versions. Hence, the inability to find the right updated information and the difficulty to share knowledge can become very costly and unproductive. According to the Meta Group, workers spend approximately 25-35% of their time searching for the information they need, rather than working on strategic projects and business opportunities. IDC Research further states that Fortune 500 companies will lose \$31.5 billion by 2003 due to rework and the inability to find information. With a unified ad-hoc P2P platform, all peers are directly connected and the most updated shared assets can be directly accessible given sufficient authorizations. In addition, a fully distributed and decentralized P2P eco-system is not plagued with some common client-server problems, such as:

- **Scalability:** A centralized computing system requires regular infrastructure upgrading to support a growing client base. Otherwise, it will become the processing, storage and bandwidth bottlenecks as the transaction traffic increases, thus leading to slow or even stalled interactivities. Centralization also presents the

* This work was supported in part by the InfoComm and InfoTech Initiative (ICITI) project grants.

possibility for a single point of failure of the entire eco-system.

- **Flexibility:** A centralized system dictates and restricts the choice of application tools for the users. All setups and configurations are managed and controlled by the administrators or service providers. While this may have some benefits, a P2P system, on the other hand, provides a natural and flexible environment for users to collaborate and share information directly with one another in an ad-hoc manner at anytime, anywhere using the most appropriate application tools.
- **Complex and High Costs:** The setup of extranets and virtual private networks (VPNs) to enable cross-enterprise collaborations encompassing multiple corporate firewall boundaries can be complex and time-consuming. It may require additional server hardware and software licensing as well as IT manpower for their operations, maintenance, and supports.

With a vision to overcome these constraints and high costs of a centralized solution for distributed ad-hoc group communications and collaboration, a number of P2P-based systems have been developed. Among these, Groove Networks [3] is one of the most established platforms in the market. However, it may not be truly scalable in an ad-hoc sense due to its reliance on some centrally managed relay servers for peer discovery and connection establishment (especially when the connecting peers are behind some firewalls). The proposed P2P ad-hoc platform based on Project JXTA aims to overcome such shortcomings.

2. THE PROPOSED AD-HOC P2P PLATFORM

Overview

Figure 1 below depicts a typical online P2P community using the proposed distributed platform. Secured ad-hoc virtual peer groups (VPG) can be easily formed for multiparty communications and knowledge collaboration among members sharing a common topic of interest.

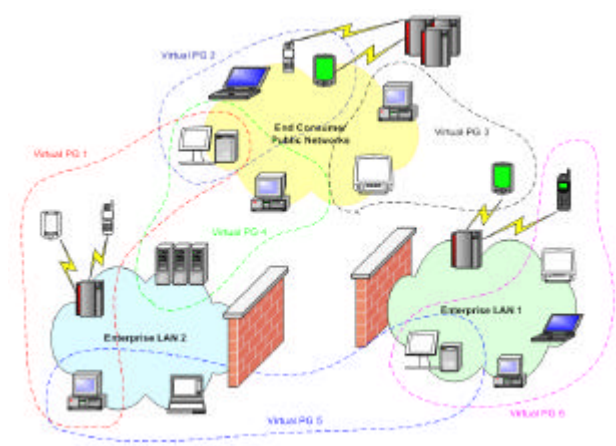


Fig. 1: A typical fully distributed and decentralized P2P eco-system without the needs and constraints of any centralized control, thus enabling the total freedom of forming secured ad-hoc virtual peer groups (VPG) comprising members sharing a common topic of interest.

Architectural Overview of Ad-Hoc P2P Platform

With the vision to develop an extensible P2P platform that supports disparate vertical applications, we have ensured that the design incorporates a good set of core APIs encapsulating the Project JXTA [7], Jabber XMPP [8], and XML Web Services [9] standards. Figure 2 shows a high-level architectural overview of the platform. It provides some core platform services such as authentication, file management and transfer, basic console GUI widgets, messaging, synchronization, multimedia streaming, etc. Third-party adapters to existing backend systems (e.g., LDAP, CRM, ERP, etc.) can also be developed, thus enabling the P2P platform to act as the middleware glue with other systems. A plethora of applications can then be developed and plugged into the proposed platform and made interoperable with other tools to create a unified and integrated peer console.

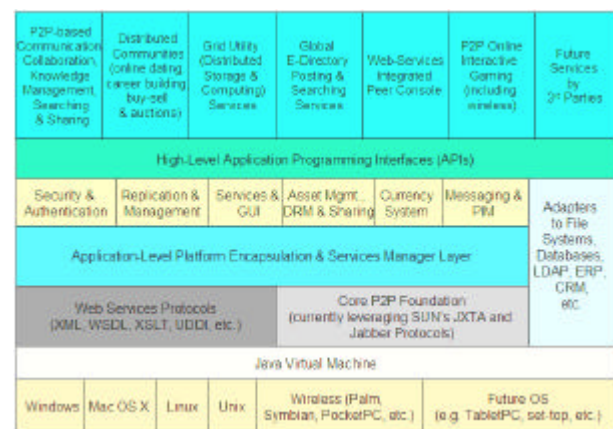


Fig. 2: Architectural overview of the proposed distributed and decentralized P2P platform. Its flexible plug-in and web services-enabled design allows easy integration of new applications and services modules via a comprehensive set of the platform APIs.

From a research and development standpoint, we focus on a number of strategic multi-disciplinary areas encompassing:

- Information security in a distributed ad-hoc environment, which includes finding better elliptic curves and developing more efficient cryptosystems; identity-based encryption and its application to simplify the deployment of public key infrastructure; data protection and access control via digital rights management; peer handshaking protocols for secure exchange of encryption keys, setup of secured peer-to-peer sessions; and localized (non-centralized) peer authentication and digital certificate/signature verification for peer identification and non-repudiation.
- Optimization algorithms for efficient global load balancing, and distributed data storage. Advanced information dispersal algorithms (IDA), which were first considered by mathematician, Michael O. Rabin [5], are also researched for developing fault-tolerant and secured data replication, transmission, searching and retrieval. Implementation of secured collaborative access via a P2P-based secret key sharing and management, and the development of feedback-free P2P communication protocols for parallel data retrieval and group data multicasting are also of great interest.

- Real-time communications and collaboration via lossy and lossless multimedia compression technologies for efficient media storage as well as online voice/video chat and messaging. This collaborative tool leverages our advanced wavelet video codec [1], [2], whereby its highly scalable compressed bit stream feature are fully exploited in a P2P environment for adaptive rate-control in multiparty conferencing over heterogeneous networks and peer devices. Scalable live video multicasting [4] and broadcasting in peer groups is also being researched for development.
- Knowledge management (KM) engine for lightweight peer-side knowledge mining processing and classification. These KM algorithms will greatly facilitate automated and intelligent indexing of information on each local peer for faster and more accurate deep searching and fast parallel data retrieval.

Handshaking Protocol for Secured Peer Session

Figure 3 illustrates one proposed peer handshaking protocol between two trusted peers who have previously authenticated with one another's digital fingerprints when they first joined the same VPG.

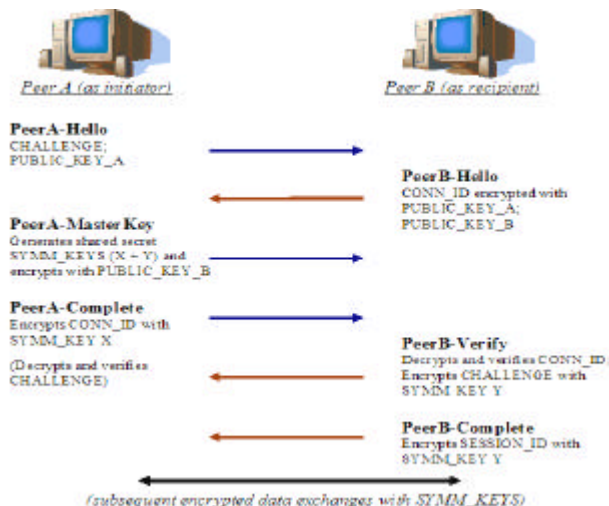


Fig. 3: Proposed peer handshaking protocol between two trusted/verified peers for secured key exchange and setup of a secured peer session using a symmetric encryption key.

The **PeerA-Hello** and **PeerB-Hello** messages negotiate a suitable cipher suite (e.g., *ECDH-ECDSA-RC4-SHA*). Peer A (the session initiator) then generates a master keys (i.e., the actual symmetric keys that will be used for encrypting subsequent data once the secured session is set up) and shares it with Peer B via **PeerA-MasterKey**. The use of CHALLENGE data and CONN_ID data are used here to circumvent possible *replay attack* by a third-party who could sniff at the channel pipes. By utilizing both Peers A and B's public keys, this protocol also prevents *man-in-the-middle attack* by forcing the engaging peers to decrypt using their respective private keys. **PeerA-Complete** and **PeerB-Verify** also serve to verify the correctness of the exchanged secret keys. Once Peer A has verified the CHALLENGE data, Peer B can then complete the handshaking via **PeerB-Complete** with the encrypted SESSION_ID. This session ID can be cached by each peer and be reused should they decide to restart a secured session within a short period of time after the termination of the previous session. This greatly speeds up the entire process by avoiding the key generation and

verification steps and reusing the already exchanged secret keys for the new secured session (if such a security level is acceptable by the application of interest).

Extension work related to identity-based encryption (IBE) [6] for public-key cryptography can also be employed. In this scenario, Alice can simply send a secure message to Bob by encrypting her message using one of Bob's unique identifiers (but publicly known to the other peers) such as the public key (e.g., bob@email.com or the unique peer ID). Now, there is *no need* for Alice to obtain Bob's public key certificate, and Bob does *not* need to pre-register with any Certificate Authority before Alice can send the secure message to Bob.

3. IMPLEMENTATION OF BASIC P2P PLATFORM APPLICATIONS

This section presents a few basic but useful application tools that are developed for the proposed P2P platform. They include ad-hoc virtual peer group (VPG) management with online peer presence monitoring using an integrated buddy roster, secured asset management and distributed sharing, searching and retrieval, synchronization of shared calendars and contacts, and scalable multimedia communications and collaboration in multiparty ad-hoc environments.

Secure Ad-Hoc Peer Grouping and Presence Management

In this P2P eco-system, a peer can create a new VPG and protect it with a password. Alternatively, the VPG can also be linked to a centralized LDAP server for membership authentication. The group owner will have full control over the membership, roles and rights of each member in the protected (can be either a public or private) VPG. In a private group, invitation to join the group can be issued to some selected peers who can then join by supplying the correct authentication credentials. Once joined, the online presence of all members within the VPG can be monitored instantly via an integrated buddy roster. As the platform also supports the standard Jabber's XMPP (Extensible Messaging and Presence Protocol), the same unified roster also displays online presence of buddies in other popular instant messaging (IM) gateways such as Yahoo!, MSN, AOL, ICQ, IRC, and Jabber. The primary focus here is to enable secured communications (both one-to-one and group chats) on all IM gateways via our unified peer consoles. Figure 4 illustrates a screenshot of the integrated buddy roster with secured chat and peer asset browsing.

Distributed Asset Sharing and Retrieval

Each member in a VPG can also conveniently manage and organize all his or her digital assets into hierarchical albums in a shareable library, and then apply specific security policies to govern the visibility, accessibility, and usage rights of the shared assets. Known types of digital assets (such as .doc, .pdf, .jpg, .gif, .mpg, .avi, .ppt, .xml, etc.) as well as the entire directory can be selected from either the peer's local hard-disk or a mapped networked drive for sharing. For example, a confidential financial document can be encrypted and protected with a password before it is made shareable with other members. The document owner can further specify the time period within which the document is available for searching and download, and the particular group of peers who can have visibility and access to the shared document.

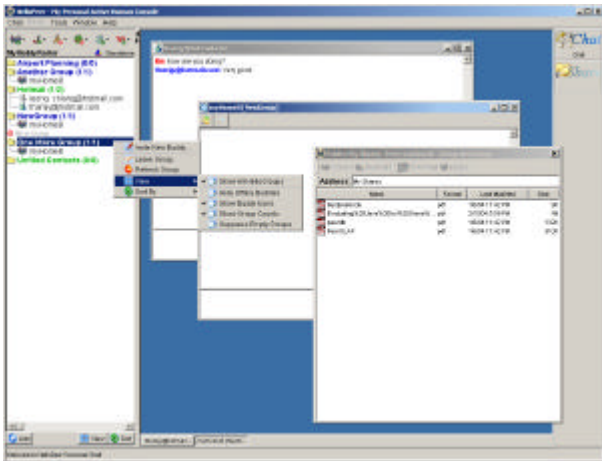


Fig. 4: A screenshot of an integrated buddy roster that provides online presence monitoring across multiple IM gateways and a unified interface for easy access to other applications, such as secured chat and peer asset browsing and retrieval.

For each shared asset, the owner can specify free-text metadata fields to describe it so that other peers with the authorized permission can search for the asset via search keywords. We have conducted research and compared two different approaches to this peer file sharing: (i) file replication and synchronization with shared asset advertisement published; and (ii) file residing on local peer and indexed locally, and searching is then performed locally too in response to distributed queries. We concluded that approach (ii) is very efficient with fast searching and greatly reduces bandwidth utilization needed for file replication. However, approach (ii) requires that the asset owner's peer console be online in order for the requesting peer to directly download the shared asset from. Extension using ideas on information dispersal algorithms for file splitting and replication on multiple peers are being considered for improved performance on parallel file retrieval and high data availability [5]. However, security protection and access control issues still require careful investigation and implementation.

Once the assets are organized for sharing, a requesting peer in a particular joined peer group can then conduct a search by specifying some keywords, asset types, and/or the selected peer groups to be searched. The search request will then be dispatched to all the relevant peers in the P2P eco-system. Each peer who receives this search query will then perform a local search (if the requesting peer has the permission) and return the search results (if any) to the requesting peer. The requesting peer will then collate and present the search results in a friendly GUI which displays some attributes of each matching asset (see Figure 5). The requesting peer may then download one or more of these files (if he/she has the permission as specified by the asset owner while creating the shared library) directly from the owner peer's device. Of course, if the owner peer is currently online, the requesting peer can also initiate a new chat session. More importantly, we ensure that all file transmissions are encrypted for secured access and content integrity checking.

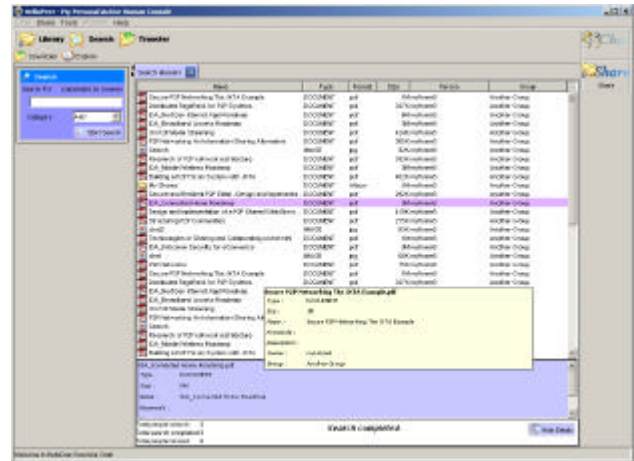


Fig. 5: A screenshot of the peer console interface which provides a simple but powerful search capability for all shared assets by other members in all joined virtual peer groups. It also allows easy asset management in a shareable library with flexible access control policies.

Synchronized Sharing of Calendars and Contacts

Each ad-hoc VPG has a common group calendar that is shared and automatically synchronized with other members in the peer group. A group calendar can also be created without being associated with a VPG. Members in a particular VPG having the appropriate rights can add, edit, delete, publish, or subscribe to the appointments in the calendar. The creator of an appointment can apply specific publication security access rules to determine the scope of visibility and editing rights of the appointment within the VPG. Once the appointments are published, other members having the appropriate access rights can (automatically or manually) subscribe for their latest updates via real-time online peer synchronization. From a graphical user interface perspective, a user can view multiple overlaid group calendars simultaneously on the peer console. Furthermore, since members in a VPG are directly connected, a meeting organizer can also browse for the time availability of each member in the group in order to schedule the best time slot for the meeting. Meeting invitations can then be sent out to all attendees and their acceptance statistics can be collated easily. Such a group scheduling capability will become very useful in such an ad-hoc group whereby the collaborating members straddle disparate corporate boundaries.

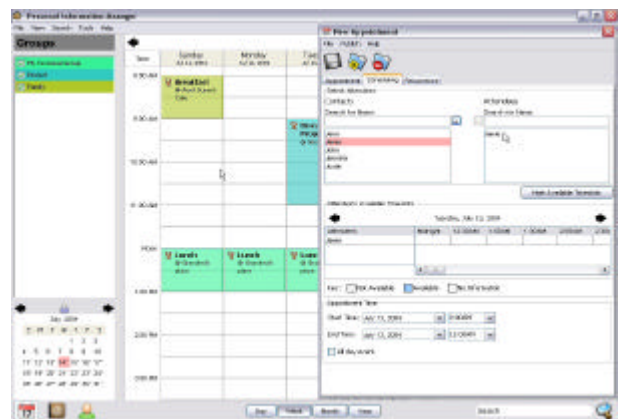


Fig. 6: A screenshot of the secured group calendar scheduling application. The meeting organizer can browse for the attendees' available time slots. Changes in the meeting appointments can then be automatically synchronized with the other members within the VPG.

In addition to scheduling group appointments, this application module also includes a shareable contacts management capability. Each peer can organize his or her contacts into multiple categories, and may choose to automatically subscribe to receive the latest updates of the contacts from the remote peers once changes have been made. The contacts owner may selectively publish only particular portions of his or her full contacts information via the creation of different virtual cards. Each virtual card is assigned with flexible publication rules to govern its subscription rights by other members. Similar to group calendaring, a peer can publish public virtual cards which can be subscribed by any peers on the P2P eco-system.

Scalable Multimedia Communications and Ad-Hoc Group Collaboration

This application module focuses on adding real-time group communications and collaboration capabilities by exploiting our highly scalable multiwavelet-based image and video compression system [1], [2]. An earlier version of the scalable video codec has been employed for the development of a layered video multicasting system to heterogeneous wired and wireless networks over the IP multicast backbone [4]. In this module, we focus on exploiting the scalable video codec for adaptive real-time video communications over an ad-hoc P2P network environment. In this implementation, different layers of the multi-scalable compressed video bit stream are multicast to multiple media sub-groups, each carrying a particular layer of the compressed video. Each peer can then selectively tune into the best possible number of media sub-groups, as limited only by the capabilities of the peer device and network connection bandwidth. In this case, the collaborating peers with different peer devices on diverse networks can still communicate seamlessly, while each peer is dynamically selecting his or her best combinations of display resolution, playback frame rate, color depth, encoding/decoding processing complexity, and streaming bit rate, which are originating from the same single source of compressed video. For example, a wireless peer may selectively tune into a low resolution and low frame rate version, while a broadband peer can receive a high-fidelity version of the scalable video multicast. Multiparty voice conferencing among peers in ad-hoc VPG is also a part of this module.



Fig. 7: A screenshot of the 3-D graphics modeling and navigation software of SGI VizServer. This tool will be integrated into the P2P platform to develop a group collaborative visualization application.

In addition to voice and video streaming, the scalable codec can also be applied for progressive image browsing of shared image libraries within the VPGs. Members having the appropriate access rights can quickly and progressively browse through a large collection of compressed image archives. During the browsing process, the peer can choose to browse a low-resolution thumbnail version of the images or to progressively stream a blur-to-fine version of the original images. With this module, peers can easily share compressed images and graphics in an ad-hoc community manner with full security access control. In addition, multiple collaborating peers within a VPG can also co-navigate and co-edit a large image together. Some useful imaging applications here include collaboration on large medical images, graphical designs and modeling, satellite images and maps, etc. Extension for collaborative real-time graphics modeling is being explored too. Here, the peers are co-navigating and interacting in real-time to render a complex 3-D model via a centralized high-performance graphics server using the SGI VizServer (see Figure 7). Depending on the collaborating peers' input commands, the VizServer will render the 3D model in real-time, and then the graphic image sequences are compressed using the scalable image codec before being delivered to the peers. Similarly, all communications are encrypted and the collaborative group session can easily be set up in a truly ad-hoc manner.

4. CONCLUSIONS

In this paper, we have proposed and developed a comprehensive and extensible peer-to-peer platform which directly connects all peers into secured ad-hoc virtual peer groups. Coupling this with different useful application tools such as integrated buddy roster and chat, secure file sharing, searching and retrieval, synchronized shared calendars and contacts, and scalable multimedia group communication and interactive collaboration, the distributed P2P platform is poised for many strategic vertical industries, such as e-Education, e-Sharing, e-Business, and e-Collaboration applications. Extensions to developing other XML web services tools such as online shopping, P2P auctioning, and universal searching via the Amazon's and Google's Web Services APIs can also be well suited for developing a more unified distributed ad-hoc P2P eco-system. Other useful tools include whiteboard sharing, survey and polling, threaded discussion groups, and bulletin boards. In a nutshell, this platform presents a practical and visionary starting point for delving into many challenging and open research problems related to P2P security, distributed and reliable databases and file storage systems, optimization of ad-hoc network routing and quality of service, multimedia streaming in ad-hoc networks, grid computing, and many more.

5. REFERENCES

- [1] J. Y. Tham, "Multiwavelets and Scalable Video Compression," Ph.D. Dissertation, Department of Electrical and Computer Engineering, National

University of Singapore, 2002. (Downloadable from:
<http://wavelet.cwaip.nus.edu.sg/thamjv/>)

- [2] J. Y. Tham, S. Ranganath, A. K. Kassim, "Highly Scalable Wavelet-Based Video Codec for Very Low Bit Rate Environment," *IEEE Journal on Selected Areas in Communications -- Special Issue on Very Low Bit-rate Video Coding*, vol. 16, no. 1, pp. 12-27, Jan. 1998.
- [3] Groove Networks at <http://www.groove.net/>
- [4] Scalable Video Multicast Over Diverse Networks at http://www.eng.nus.edu.sg/EResnews/1003/rd/rd_9.html
- [5] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," *Journal of the ACM*, vol. 36, no. 2, pp. 335-348, April 1989.
- [6] D. Boneh, and M. Franklin, "Identity based encryption from the Weil pairing," *SIAM Journal of Computing*, vol. 32, no. 3, pp. 586-615, 2003.
- [7] Project JXTA, Sun Microsystems at <http://www.jxta.org/>
- [8] Jabber XMPP Protocols at <http://www.jabber.org/protocol/>
- [9] XML Web Services Standards at <http://www.w3.org/2002/ws/>



Tham Jo-Yew is currently a Research Fellow with the Centre for Industrial Mathematics, National University of Singapore (NUS). He was a recipient of the ASEAN scholarship, and obtained his bachelor degree and Ph.D. in Electrical and Computer Engineering, NUS, in 1995 and 2002 respectively. Between 1996 and 1999, he was a Research Associate

with the NUS Centre for Wavelets, Approximation and Information Processing. He later spent three years with two multimedia high-tech startup companies in the Silicon Valley, USA, as Senior R&D Manager and Chief Streaming Architect/Chief Technology Officer. His current research focus includes distributed peer-to-peer and grid computing, distributed information security infrastructure, scalable multimedia compression, group collaboration, and knowledge mining, sharing and rights protection. He is also a member of MENSA and IEEE.

The Splitting Methods in High-speed Networks Data Analysis

Chen Xunxun¹ Fang Binxing² Li Lei³

Dept. of Computer Science, Harbin Institute of Technology

Harbin, Heilongjiang Province 150001, P. R. China

Email: {cxx¹, bxfang², lilei³}@pact518.hit.edu.cn Tel: 13911226679

ABSTRACT

We put forward three splitting methods in high-speed networks data analysis, which are traditional IP routing method, streams classifying based on policy routing and streams classifying based on hashing. These techniques can be used to solve the bottleneck problem of the performance in large-scale backbone networks data analysis. The result is made by continuously sampling experiments data on real networks. By slightly modified, the streams classifying method, which is designed for supporting QoS service, can be a high performance, lightweight, stable, clustered and mostly non-bursting splitting technique. This technique has high scalability and flexibility and can be improved on efficiency, complexity when combined with hashing technique.

Keywords: Splitting, Load Balance, Streams classifying, Networks Data Analysis, High-Speed Networks, Hashing

1. INTRODUCTION

At present the number of Internet users in China has occupied the second position among all countries in the world [1]. With the growing of networks scale, the increasing of bandwidth, the improvement of information technology and the explosive of number of Internet users, the high-speed networks become more and more popular. And thus comes with a lot of security problems. Intrusion detection systems (i.e. IDS) [2] and other content analysis software are important and practical techniques for solving nowadays network security problems. We define the high-speed networks as networks that have more than one 2.5Gbps links in their backbone network. IDS and other content analysis software are normally mounted on the core switches and the gateways of the network. Now the study on IDS encounters a special problem that is the challenge for data processing speed. The famous information security research and advising enterprise, Gartner Inc., puts forward a contention that IDS will die away by 2005 [3]. One of its four reasons is that the current IDS cannot handle real-time network data, the speed of which can often exceed 600Mbps.

The first problem that should be solved for IDS and other content analysis software on high-speed networks is how to balance the high bandwidth data reasonable so that it can fit the performance of the analyzing machines. In high-speed IDS, a splitting device is designed for such motivation normally [4] [5] [6] (see figure 1 also). Raw network packets are captured and transferred to a splitting device through certain taps and then these packets will be gathered and scattered to sensor nodes after load balancing. However, at present journal articles or books written for splitting technical or its implementation are hard to be found in the world [7] [8].

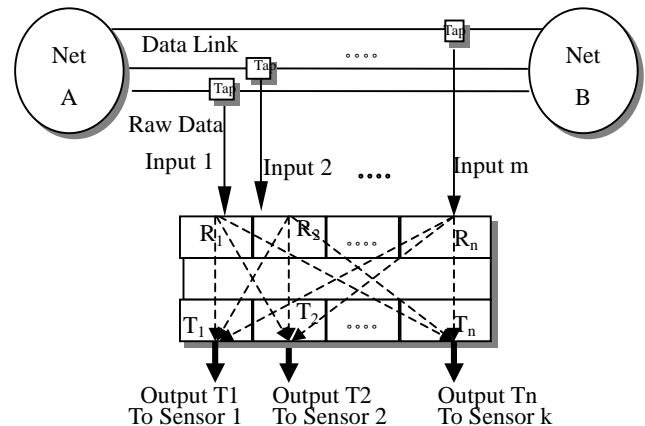


Figure 1 . Front-end Architecture of Content Analysis Systems on High-speed Networks

For finding the suspicious operations in raw packets of network data, the content analysis systems like IDS should look through the payload of application layer in the data streams [9] [10] which means that all data in a certain data stream must be forwarded to the same analyzing node. So the splitting device must perform load balancing in TCP layer [5]. The standard Round-Robin load balancing methods in traditional routers cannot fit such requirement, whether by IP or by packets. At the same time, because splitting devices must have powerful forwarding ability for keeping up with core routers' performance, traditional load balancing methods, which have down-stream nodes state checking function and are more complex, more time-consuming and not easy to implement by hardware, also cannot fit the requirement of real-time load balancing in TCP layer. Load balancing systems for network content analysis systems on high-speed networks (i.e. splitting systems, distinguished from traditional accept of load balancing) need to fit three special requirements as follows: First, Same Source Same Destination (SSSD) that is forwarding the two-way data of a TCP/UDP stream to a same sensor; Second, Dynamically Balancing (DB); And third, High Speed and Light Weight (HSLW). At present, two series of splitting methods are often used in practice. One is traditional routing method and the other is streams classifying method that can be further divided into streams classifying based on policy routing and streams classifying based on hashing function.

2. TRADITIONAL ROUTING METHOD

Definition: E is the set of all IPv4 addresses; A is the set of IPv4 addresses in network A; B is the set of IPv4 addresses in network B; Then $A \cap B = E \setminus A \setminus B$. Make a division of A: $A_1, A_2, \dots, A_n, n \in \mathbb{I}^+$; Then $A_1 \cup A_2 \cup \dots \cup A_n = A, A_i \cap A_j = \emptyset, i=1..n, j=1..n, i \neq j$. Make a division of B too: $B_1, B_2, \dots, B_m,$

$m \quad I^+$; Then $B_1 \quad B_2 \quad \dots \quad B_n=B, B_i \quad B_j= \quad i=1..m \quad j=1..m$,
 $i \quad j$. Let $E_i=A_i, i=1..n, E_{i+j}=B_j, j=1..m$, then E_1, E_2, \dots, E_{n+m} ,
 $m, n \quad I^+$, is a division of E .

Splitting policy based on traditional routing method constructs a routing table and forwards the input packets to $m+n$ output ports according to their destination IP addresses range (from E_1, E_2, \dots , to E_{n+m}). It is easy to find that each processing node behind the splitting device has only one-way packets of a certain TCP stream. It increases the complexity and relationship of the processing and analyzing software. And the analysis result of one node will be modified by other nodes' running state. On splitting performance, this method has an advantage that normal configured routing device can be used as splitting device which can be available easily. Especially in high-speed networks, commercial high-end routers can take the role of splitting device. On the other hand, the output flows rate changes in a wide range and bursting flows often occur on the statistics chart, and the σ^2 is also not good, especially in low-speed networks. At the worst time, the output rate can easily exceed the bandwidth of output ports or the ability of processing nodes behind the output ports and thus a large number of packets are lost or nodes are congested. Splitting based on traditional routing method can hardly balance the load among the output ports, but the output rate can be stable in most time except for the bursting data. So high-end routers can implement this method and more bandwidth should be reserved for output ports. Figure 2 shows the statistics chart of a splitting system has eight POS2.5G input lines and sixteen GE output ports.

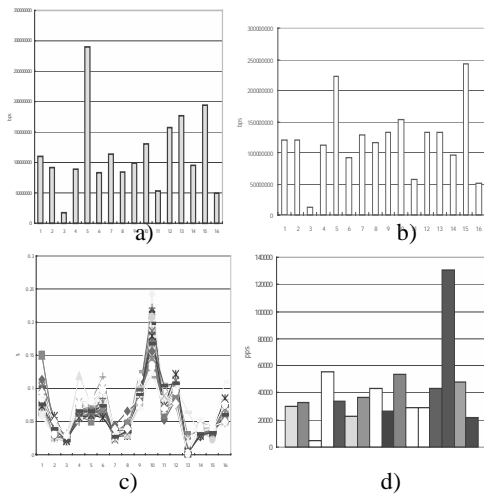


Figure 2 . Statistics Chart of Splitting Policy Based On Traditional Routing

By analyzing one-month observation data on a high-speed network, the bandwidth of which is more than 5Gbps, we get the result in figure 2 and table 1. Chart 2a and chart 2b show flow rates on all output ports observed in the first day and the thirty-seventh day. It can be found that they are not different very much. Chart 2c shows the ratio of the flow rate of each port in total rate and it is very stable in most time. Chart 2d shows the flow rates of the sixteen output ports in another day when emergency events occur in the network, and that day is not included in the thirty-seven days. It shows that one of the output ports has a clearly bursting flow.

Table 1 Statistics Data of Splitting Policy Based On Traditional Routing

Items	Mean of Proportion of Total Rate	σ^2	Coefficient of Variation	Range / Mean	Quartile Deviation / Mean
Output 1	8.15%	0.000463	26.78%	114.88%	23.59%
Output2	3.37%	0.000061	23.54%	101.54%	18.88%
Output3	2.48%	0.000007	11.13%	48.12%	14.49%
Output4	6.64%	0.000130	17.39%	106.32%	18.90%
Output5	6.96%	0.000061	11.38%	51.13%	17.36%
Output6	7.75%	0.000222	19.52%	80.08%	24.52%
Output7	3.00%	0.000020	15.27%	83.10%	13.62%
Output8	3.83%	0.000069	22.01%	102.98%	18.05%
Output9	9.01%	0.000144	13.52%	63.22%	12.01%
Output10	17.28%	0.000913	17.73%	72.25%	22.86%
Output11	7.35%	0.000135	16.03%	73.07%	14.16%
Output12	9.80%	0.000177	13.77%	50.85%	23.31%
Output13	0.59%	0.000118	185.38%	1037.53%	29.88%
Output14	3.69%	0.000032	15.48%	65.01%	21.55%
Output15	3.66%	0.000026	14.13%	70.74%	12.21%
Output16	6.43%	0.000156	19.73%	101.01%	24.91%
Mean	6.25%	0.002734	27.67%	138.86%	19.39%

According to the statistics data we can conclude the typical laws of splitting systems based on traditional routing method. After splitting rules are established, the splitting ratio of each output port is generally stable and the flow rate of a certain port changes smoothly in most time too except for the concurrent bursting flows (see also the range/mean data of port 1, 4, 8, 13 and 16). The ranges of the bursting flows change greatly (see also the range/mean data of port 1,4,8,13 and 16) and last for a short time (see also the range/mean and the quartile deviation/mean data of port 1,4,8,13 and 16). The splitting ratio of this policy is generally stable, but the problem coming with the bursting flows should be handled especially on small splitting ratio output ports. By further analyzing on the bursting flows, we can find these flows consist mostly of DDoS packets. However, because the splitting ratio of the thirteenth port is very low, its peak ratio (approx. 6.21%) is still below the mean splitting ratio (6.25%). So the processing node behind it can still afford the flows in practice. We can conclude that splitting systems based on traditional routing method can be implemented easily and can efficiently perform the splitting function. The most serious limitation of this technique is that it cannot fit the requirement of SSSD and it increases the complexity of back-end systems.

3. STREAMS CLASSIFYING BASED ON POLICY ROUTING (SCBPR)

Because traditional routing method splits packets only according to the destination IP address, it cannot achieve the SSSD aim. The SSSD aim can be achieved only by splitting on TCP streams that is protocol field, source IP address field, destination IP address field, source port field and destination port field of every packet being seen. This can also be implemented by using high-end routers support streams classifying based on policy routing. The definitions of set E , A and B are same as in section 1. We define a new set C based on the definition in section 1: $C: \{ \langle p_i, sip_p, spt_k, dip_x, dpt_y \rangle \}$, elements in which are tuple5s, and p_i is protocol, sip_j is source

IP address, spt_k is source port, dip_x is destination IP address, and dpt_y is destination port. According to the definition of IPv4, we have $p_i=0.2^{16}-1$, $sip_j=0.2^{32}-1$, $spt_k=0.2^{16}-1$, $dip_x=0.2^{32}-1$, and $dpt_y=0.2^{16}-1$. This splitting policy is constructing a policy routing forwarding table and forwarding the packets that have $\langle p_i, sip_j, spt_k, dip_x, dpt_y \rangle$ and the packets have $\langle p_i, dip_x, dpt_y, sip_j, spt_k \rangle$ to the same output port. Theoretically, to have the smallest size of splitting granularity, it needs to construct a table including $2^{4194304}$ elements according to the range of each element of set C, but we see that this way is impossible to implement in practice. Fortunately we can use masks to make an approximately balanced splitting according to the requirement of the size of the splitting granularity. Define two masks Mip , the size of which is 32 bits, and $Mport$, having a 16-bit size. Then mask sip and dip with Mip and mask dpt and mpt with $Mport$ to construct a new set $C':\{\langle p_i, sip_j', spt_k', dip_x', dpt_y' \rangle\}$, the size of which is determined by the number of binary '1' in Mip and Mpt . For example, if only TCP and UDP packets are valid and let $Mip=1$ and $Mpt=1$, then the number of streams classifying rules is $2^5=32$. And the number would be $2^9=512$ if $Mip=3$ and $Mpt=3$. IP addresses are generally assigned from left to right, so with a small rules number it will be more efficient when right bits of the addresses are left after mask operating. Figure 3 shows the statistics of the flows of the eight output ports of a policy routing based streams classifying splitting system on a high-speed network with two GE lines and two FE lines. The valid packets are TCP and UDP packets with Mip equals to three and Mpt equals to three too.

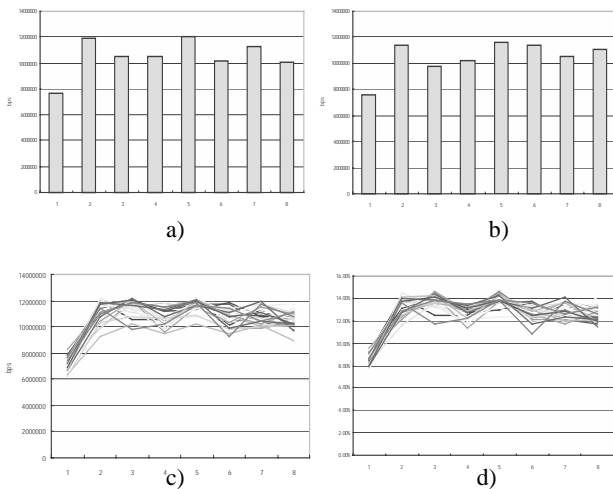


Figure 3. Statistics Chart of Splitting Policy of Policy Routing Based on Streams classifying

From one-month observation data on a high-speed network, the bandwidth of which is greater than 2Gbps, we get the result in figure 3 and table 2. Chart 3a and 3b show flow rates (bps) on all output ports observed in the first day and the thirty-first day. It can be found that they are not different very much. Chart 3c and chart 3d show the flow rate and the ratio of flow rate of each port in total rate, and they are very stable all the time. The bursting flows do not occur at all.

According to the statistics data we can conclude the typical laws of splitting systems based on policy routing streams classifying. After splitting rules are established, the splitting ratio of each output port is very stable and average, and the flow rate of a certain port changes smoothly all the time

Table 2. Statistics Data of Splitting Policy of Policy Routing Based On Streams classifying

Items	Mean of Proportion of Total Rate	2	Coefficient of Variation	Range / Mean	Quartile Deviation / Mean
Output	8.53%	0.000030	6.58%	22.46%	12.37%
Output	13.22%	0.000048	5.35%	22.40%	7.54%
Output	13.69%	0.000034	4.30%	20.97%	4.52%
Output	12.71%	0.000038	4.94%	18.80%	7.68%
Output	13.78%	0.000021	3.42%	16.38%	4.00%
Output	12.67%	0.000039	5.01%	22.65%	5.93%
Output	12.79%	0.000044	5.26%	18.44%	8.40%
Output	12.60%	0.000037	4.89%	20.98%	7.03%
Mean	12.50%	0.000291	4.97%	20.39%	7.18%

without any bursting flows being found. We can see that streams classifying method is much better than traditional routing method on splitting ratio balancing, stability and coefficient of variation and it can achieve the SSSD aim. The disadvantage of streams classifying method is that the configure rules are complex and only little high-end switch-routing device can support line-speed and non-block streams classifying on high-speed networks.

4 STREAMS CLASSIFYING BASED ON HASHING (SCBH)

The problem of the complexity and difficulty of rules configuration adjusting in policy routing based streams classifying can be solved by hashing technique. This method uses a hashing function, which is described as $H(Sip, Dip, Spt, Dpt)$, where Sip , Dip , Spt and Dpt are source IP address, destination IP address, source port and destination port of a IP packet p . The function $H(p)$ must meet the requirement as follows:

$$H(Sip, Dip, Spt, Dpt) = H(Dip, Sip, Dpt, Spt)$$

If we have n output ports, the output port number Tn of a received packet can be calculated by the formula as follows:

$$Tn = H(Sip, Dip, Spt, Dpt) \bmod (n + 1)$$

The more $H(p)/n$ is big, the more the splitting granularity becomes small. And with the bigger $H(p)/n$, the splitting ratio is more even and the ability of anti-bursting is more powerful. The reason is that when large clustered flows occur (e.g. access to large sites like Google.com or high-speed proxies), the source ports of the TCP streams are various (incrementally for most client PC) though the IP addresses of the streams are same. Because hashing technique turns logical operation to arithmetic operation that can highly improve the operating speed, processing real-time network data in line-speed on very large bandwidth networks becomes available. The function and performance of policy routing method are same as those of hashing method, so that statistics data are not given again.

5. CONCLUSIONS

We put forward three splitting methods in high-speed networks data analysis, which are traditional IP routing method, streams classifying based on policy routing and streams classifying based on hashing. These techniques can be used to solve the bottleneck problem of the performance in large-scale backbone networks data analysis. The result is

made by continuously sampling experiments data on real networks. Only by slightly modified, the streams classifying method, which is designed for supporting QoS service, can be a high performance, lightweight, stable, clustered and mostly non-bursting splitting technique. This technique has high scalability and flexibility and can be improved on efficiency, complexity when combined with hashing technique.

Table 3. The Contrast of the Two Lightweight Splitting Methods

		Coefficient of Variation	Range / Mean	Quartile Deviation / Mean
Traditional Routing Based	2.38%	27.67%	138.86%	19.39%
Streams classifying Based	1.67%	4.97%	20.39%	7.18%



Fang Binxing is a professor of School of Computer Science and Technology, Harbin Institute of Technology. He is also part-time professor of Institute of Computing Technology, Chinese Academy of Sciences. He is a director member of Network and Information Security Committee of China Institute Of Communications. He is also a deputy director member of Professional Committee of Computer Security of China Computer Federation. He is major on computer network and information security, computer architecture and parallel computing.

6. REFERENCES

- [1]. <http://www.cnnic.org.cn/html/Dir/2003/11/04/1203.htm> , Mar. 2004
- [2]. Dorothy E. Denning, An Intrusion-Detection Model, IEEE Transactions on Software Engineering, Vol. SE-13, No. 2, Feb. 1987, 222-232.
- [3]. http://www3.gartner.com/5_about/press_releases/pr11june2003c.jsp, Jun. 2003.
- [4]. J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, E. Stoner, State of the Practice of Intrusion Detection Technologies, Carnegie Mellon University/Software Engineering Institute Technical Report CMU/SEI-99-TR-028, Jan. 2000.
- [5]. R.Sekar, Y.Guang, et. al., A High-Performance Network Intrusion Detection System, 1999
- [6]. Christopher Kruegel, et. al., Stateful Intrusion Detection for High-Speed Networks, 2002.
- [7]. Richard Lippmann, et. al., the 1999 DARPA Off-Line Intrusion Detection Evaluation, 2000.
- [8]. D. Song, G. Shaffer, M. Undy, Nidsbench – A Network Intrusion Detection System Test Suite, Second International Workshop on Recent Advances in Intrusion Detection(RAID), Sep. 1999.
- [9]. William W. Cohen, Fast Effective Rule Induction, Machine Learning: Proceedings of the Twelfth International Conference, 1995.
- [10]. R.Bettati, W.Zhao, D.Teodor, Real-Time Intrusion Detection and Suppression, Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, CA, April 1999.

AUTHOR CURRICULUM VITAE



Chen Xunxun is a doctorate candidate of Research Center of Computer Network and Information Security Technology, Harbin Institute of Technology. He's major on computer network and information security.

Design and Implementation of Policy-based Network Management Based on SNMPv3*

Yi Yue, Debao Xiao

Computer Science Department, Central China Normal University
Wuhan, HuBei, China

Email: ccnuyy3218@163.com Tel: +86 (0)27 67866108

ABSTRACT

This paper introduces the concept and model of PBNM based on business rules for the start. Following it, the implementation framework of SNMPv3 is given. Based on it, the implementation framework of PBNM is given at the end of this paper.

Keywords: SNMPv3, PBNM, Business Rules

1. INTRODUCTION

The framework of network management mainly based on network elements in the past^[1] the administrators required to know all the management interfaces in the Managed Object, and have a good command of the values of MIB (Management Information Base) variables. They have many dull jobs to monitor the status of running network and to configure the parameters. All these jobs are very boring. The administrators' working abilities have a direct effect on the validity of network management. With the scale development and of network, there are more and more network devices and users. A majority of network devices such as host, server, switch, router and hub etc come from different manufacturers. This situation that leads network to be more huge, complicate and heterogeneous aggravates administrators' jobs. It is known as the bottleneck of network management. Further more, with the application of multimedia in the network, the network is becoming more congested, and users demand higher QoS (Quality of Service). Aim to solve these tasks in the network management, here comes a new solution—Policy-based network management (PBNM)^[2], and it is becoming a rapid developing technology in network management.

On the another hand, with the development of computer network technology, the scale and application of network is becoming more complicate, how to successful manage network and to improve performance and QoS is becoming an imminence task. The Internet Organization released SNMP (Simple Network Management Protocol)^[3] for network management. This protocol is accepted by many manufacturers and becomes an actual standard because of its simpleness and convenience manipulation.

This paper presented the design and implementation PBNM based on SNMPv3 that extended SNMPv3 applied to network management.

2. POLICY-BASED NETWORK MANAGEMENT

2.1 Policy Concept

The policy is a set of plans that an organization constitutes to realize its aims in a general scope, but it is a set of rules that direct how to manage, distribute and control network resources in the network management scope^[4]. In fact, the concept of Policy is come from business scope, and it only presents business rules or goals. Therefore business rules are the conditions of the action.

IETF (Internet Engineering Task Force) plays an important role in the development of PBNM. IETF and DMTF (Distributed Management Task Force) defined Policy Core Information Model^[5] and its extension version, and they also put forward implementation framework of network management system based on policy^[6]. Based on CIM (Common Information Model), IETF defined policy model based on Condition-Action rules, and there are five components in the policy network management schema. These five components are Policy Repository, Policy Editor, PDP (Policy Decision Point), PEP (Policy Enforcement) and Policy Monitor. The user edits policy by used Policy Editor. The Editor stored the policy after processing it. PDP translates policy rules and launch the action. PEP obtained the action command and executed it and the Policy Monitor monitors the result. In addition, COPS (Common Open Policy Service) is defined by IETF, and it extended Policy Core Information Model in the scope of QoS and network security.

2.2 Policy-Based Network Management Model.

According to the concept of policy mentioned above, policy actually defined the action that a system wanted to execute or the status that it expected to achieve. Policy itself can be classified from higher organization direct policy to lower program code policy by several abstract levels. All these levels are combined into a level architecture of policy. The policy of network management is in a relative lower level in this paper. It defined policy as a set of ECA (Event-Condition-Action) rules, and when event A is triggered and condition C is meted, action A is executed.

The policy is a function mapped from one event to a set of actions based on the above analysis. PBNM system can be seen as a state machine. The policy decided the valid states of

* This work was supported by the Provincial Great Science Project of HuBei (grant 2001AA104A05) and the Provincial Science Foundation of HuBei (grant 2001ABB013).

managed devices. The event and condition in the policy rules defined the environment that policy applied, and they can be denoted as a disjunction or conjectured normal formula of condition proposition. If the condition is true, the action set is executed. This can lead to the system stay at former state or transfer a new state. The policy may be one rule or a set of rules, and the policy set can be nested each other. Figure 1 show this situation.



Figure 1 Policy and Policy rules model

Based on different triggering condition, policy can be classified as two patterns: static and dynamic pattern. Static policy uses a fixed action set in term of pre-defined parameters. For instance, VOD transaction can be used only after five o'clock p.m., and some IP address users are forbidden in some network resources for network security. And that dynamic policy is executed only it is needed.

2.3 The Framework of PBNM System

In general, there are at least three basic function components when implementing PBNM system.

- Policy Repository—stores and search policy. For instance, it can be a relational database. The directory server is popular accepted at present.
- PDP—is a entity that decided action based on policy rules and network service state, and it can translate policy rules and launch responding action. For instance, after PDP accepted request and policy condition from PEP, it matches corresponding policy in Policy Repository and return action to PEP.
- PEP—is a entity that actually executed policy action. After sending policy decision request and providing policy condition, it translated policy decision into configuring operation commands related to some devices and executed it.

To make the administrator create and manage policy convenience in a system, a Policy Editor and Policy Monitor is needed. The user interfaces that have functions of editing, browsing and validating are provided by Policy Editor. And they can convert natural language policy that the administrator inputted into a set of rules and store it in Policy Repository. Policy Monitor is a function that receives feedback information. When a lot of network state information is sent to Policy Monitor, it checks the effect of executed policy. Meanwhile, it can adjust the action into a prospective aim by the action of PDP.

Based on the fact mentioned above, a framework^[7] of network management system consisted of PEP, PEP, Policy Repository, Policy Editor and Policy Monitor as Figure 2 showing. For new devices that support PBNM, PEP software can be embedded in these devices. But an external PEP agent is needed for traditional devices in a PBNM system. After the PEP agent transferred the policy sent by PDP into corresponding commands related to management interfaces of some devices, the device can exchange information readily. A new communication protocol is needed when information transmitted between PDP and PEP, for example COPS. The

LDAP^[8] can be used when Policy Repository accessed by PDP. Policy Monitor supervised the state of network in real time and return result to PDP. The PDP, PEP, Policy Monitor constitutes a management activity circle.



Figure 2 A Framework of PBNM System

3. SNMPV3-BASED NETWORK MANAGEMENT

3.1 SNMPv3 Management Overview

SNMP is an application protocol based on TCP/IP. It uses UDP as transport protocol and can manage network devices supported by the proxy. SNMP is consisted of SMI (Structure of Management Information), MIB (Management Information Base) and SNMP protocol.

SNMPv3, the third version of SNMP, is different from first and second version. It presents new management architecture. The SNMP manager and agent is all called as SNMP entity in this version. An entity comprises an SNMP engine and a set of applications^[9]. The SNMP engine, named snmpEngineID, consisted of a dispatcher, a message processing subsystem, a security subsystem and access control subsystem.

- Dispatcher performs three sets of functions. First, it sends messages to and receives message from the network. Second, it determines the version of the message and interacts with the corresponding message processing model. Third, it provides an abstract interface to SNMP applications to deliver an incoming PDU to the local application and to send a PDU from the local application to a remote entity.
- The Message Processing Subsystem. The SNMP message processing subsystem of an SNMP engine interacts with the dispatcher to handle version-specific SNMP messages. It contains one or more message processing models. The version is identified by the version field in the header.
- The Security and Access Control Subsystems. The security subsystem provides authentication and privacy protection at the message level. The access control subsystem provides access authorization security.

3.2 Implementation Framework of SNMPv3

RFC3411 presents the SNMPv3 architecture. Some changes are needed when the SNMPv3-based network management platform is implemented in our laboratory. On the premise of warranting the security service of SNMPv3, the platform should be simplified as possible. Therefore, the following Figure 3 is the implementation framework of SNMPv3 in the practice. The dashed box at the top of level presents the general SNMP API used by users. The SNMP entity is presented as the bold box under it. This part contains SNMP

operation primitive module and SNMP engine model, and the former is consisted of Command Generator, Notification Originator and Notification Receiver applications (The VACM is needed for the Agent). The SNMP engine is consist of Dispatcher, Message Processing Subsystem and Security Subsystem. The bottom of SNMP entity is the part of transmitting network that mainly used UDP or TCP protocol. Each subsystem or model exchanges information using messages in this framework. Only in one subsystem or model, the message can be disassembled and turn into corresponding processing part.

4. THE IMPLEMENTATION FRAMEWORK OF SNMPV3-BASED PBNM

Because of security aspects in SNMPv3 network management, for authentication and privacy protection, the policy-based management and SNMPv3 should be taken into account in the practice. In addition, how to translate the policy rules into the corresponding SNMP primitives, and return the results to PDP is the main problem. On the basis of all the facts mentioned above and the implementation architecture^[10] that Jenne Wong presented at New Zealand Canterbury University, Figure 4 is our SNMPv3-based PBNM architecture. PEP translates the policy sent by PDP into corresponding network management commands. It can be a single PEP agent, or as a part embedded into management agent. PDP decides corresponding policy according to rules and network services, and searches matched policies from directory server using LDAP at the same time. At last, PEP obtains the results. PDP accepted the request and policy condition from PEP in order to adjust policy action. For instance, the device may send Trap primitive to the manager, and PEP must receive this Trap at first. After extracted the information available, PEP returns the request to PDP. In Figure 4, the Manager that is a policy manager in a general scope is different from the traditional manager. There are many Managers in the managed network, and they cooperate with other Managers.

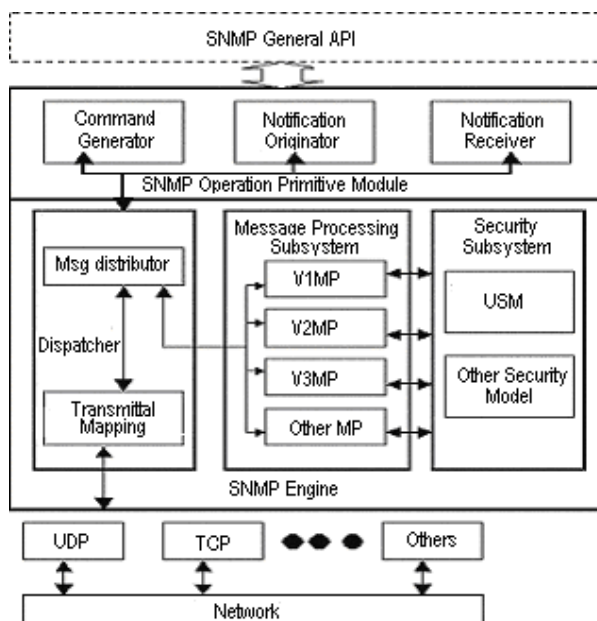


Figure 3 Implementation Framework of SNMPv3

corresponding security model at GUI interface. The advantage of this is PEP can configure security parameters of network management commands when translating policy into corresponding commands. For instance, the necessary authentication and privacy protection is needed for the message. Moreover, PEP may use the first or second version of SNMP to create message for managing network if the task has higher priorities to execute or lower security level.

5. CONCLUSIONS

The schema that SNMPv3-based PBNM combines PBNM and SNMP takes the advantages of business management of PBNM based on SNMP. Meanwhile, it improves the efficiency of managing network and makes the network administrators adopt rules to manage network. Following it, the business one set of rules is translated into device instructions. This schema reduces the faults originated from the device-central traditional management technique and improves the retractility. The network administrator will focus on business requirement not on details of device configuration. We proved this advantage by a large amount of experience in developing and applying network management system in practice.

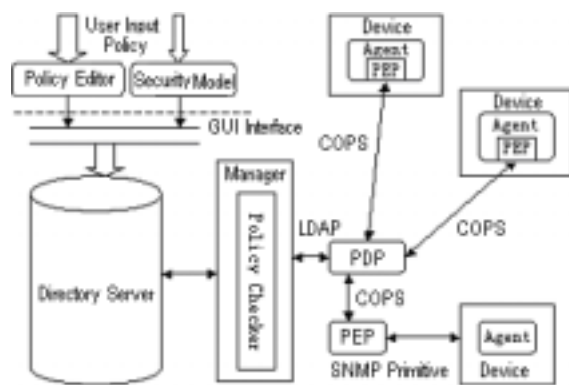


Figure 4 The Framework of SNMPv3-Based PBNM

6 REFERENCES:

- [1] Cen Xiandao, An Changqing The network management protocol and its Applications Tsinghua University Press 1998.7 (in Chinese)
- [2] Zhao Wei, Shao Junli Policy-based network management Academic Journal of PLA University of science and technology (2001-No.3). (in Chinese)
- [3] Guo Xueli, Feng Juming etc. SNMPv3-Higher Security of network management protocol (2001-No.1) (in Chinese)
- [4] R YAVATKAR, D PENDARAKIS, R GUERIN. A framework for policy-based admission control [EB/OL]. Draft-ietf-framework-01.txt,IETF, <http://www.ietf.org>.1998.
- [5] RFC 3060-2001, Policy core information model-version1 specification [S].
- [6] STEVENS M, WEISS W, MAHON H, et al. Policy framework [EB/OL]. Internet-Draft, 1999-09.
- [7] Song Lihua, Wang Haitao. The framework of PBNM. China Data Communication (2002-No.11) (in Chinese)
- [8] RFC1777—1995 Light weight directory access protocol [S].
- [9] [US] David Zeltserman SNMPv3 and Network Management Ren Min Post Press 2000.5
- [10] Jenne Wong Policy Based Network Management IIR Conference, London, Feb. 2001

The user inputs management policy and chooses

Implementations of CGI in Embedded Web Server

Xihuang Zhang, Wenbo Xu
School of Information Engineering, Southern Yangtze University
Wuxi, JiangSu 214036, P.R. China
Email: zxh163com@163.com Tel.:+86-0510-8876518

ABSTRACT

Embedded web server now is applied in many fields especially in industry control devices and CGI is the basic approach by which the client device can communicate with the web server through a simple way. In this paper, a new designed CGI in the embedded Web server based on MCU systems without operating systems is discussed, and CGI become the only useable access to interact with the Web server in embedded device. This article shows how the implementations and design of CGI in the embedded environments, and the main functions and user API also are introduced here.

Keywords: CGI, embedded Web server, embedded environments, MCU

1. INTRODUCTION

Web browsers have become the standard user interface to a variety of applications. They can run on almost any platform—from PCs and workstations to PDAs, cell phones, and pagers—and allow end-users to access Web-enabled applications from any location. Today, Web technology is becoming a key role in our daily life. Web servers and Web pages are definitely the things general people meet most. It is most convenient for users to control remote intelligent appliances or other devices with a browser connecting to the Internet. Embedding a 'tiny' Web server in these home appliances is the trend of current computer technology. Allow for there have millions of 8-bit or 16-bit MCU been used in control devices, and the cost of 8-bit or 16-bit MCU is significantly lower than that of 32-bit systems. So, if we can implement the Web server suitable to 8-bit or 16-bit MCU environment, it will accelerate the progress to let 'isolated' low-end control devices become interconnected. However, there are some disadvantages of 8-bit or 16-bit MCU that make developing Web servers in them very difficult. Such as small memory size (i.e. no more than hundreds of Kbytes), few reliable and secure operating systems fit for them, without file systems etc. So, the program needs memory management, file management and process scheduling cannot run in 8-bit or 16-bit environments.

It is well known that CGI is the basic mechanism that people can use to interact with Web servers. Users can definitely monitor remote devices with embedded Web servers supporting CGI functionality. But standard CGI needs file management and process scheduling. So, standard CGI cannot be implemented in this sort of low-end embedded Web server.

In this paper, we propose a novel approach to implement the CGI in embedded environments. And the main algorithm is discussed in the following paragraph.

2. THE THEORY OF CGI

2.1 Basic concepts

One possible solution is modeled after the CGI found in many traditional Web servers. In this model, each URL is mapped to a CGI (Common Gateway Interface) script that generates the Web page. In a typical embedded system, the script would actually be implemented by a function call to the embedded application. The application could then send raw HTML, XML, or other types of data to the browser by using an interface provided by the embedded Web server software. CGI defines the interface standard for Web servers to communicate with CGI scripts. In Web environments, the browser sends some information to the Web server. The Web server puts the needed information into environment variables then invokes a special CGI program to process them. The CGI program fetches information from environment variables to process and returns the results in HTML format to the browser via the Web server. Because Users can send variable parameters to the CGI program, the CGI technique offers the interaction between Web servers and Web browsers.

2.2 The CGI Data Transferring

CGI main functions transmit some information from client side to server side through Internet. One possible solution is modeled after the CGI found in many traditional Web servers. In this model, each URL is mapped to a CGI script that generates the Web page. In a typical embedded system, the script would actually be implemented by a function call to the embedded application. The application could then send raw HTML, XML, or other types of data to the browser by using an interface provided by the embedded Web server software. CGI uses the HTML form to send data to Web servers. The simplest syntax is given as follows:

`<FORM METHOD=get/post ACTION=URL ></FORM>`
'METHOD' attribute denotes the method to send data to Web servers. There are two basic methods in used: In 'GET' method, the data set in the form will be sent to the Web server as a postfix of the CGI program's URL. The Web server then puts the received form data into the environment variable 'QUERY_STRING'. And in 'POST' method, form data are sent to the Web server through 'STDIN'. Then the Web server invokes the CGI program to receive the data from 'STDIN'.

'ACTION' attribute denotes the URL of the CGI program to process form data.

The output method of CGI sends the results to STDOUT. The result must be processed into HTML format.

During the entire procedure of data transferring, some parameters need to be sent to the CGI program via the Web server. To achieve this goal, Web servers need environment variables to store the parameters received from the client.

Then the CGI program can obtain these parameters from environment variables. In other words, Environment variable is the mechanism to transfer data from the Web server to the CGI program.

2.3 CGI Flow Chart

Figure 1 is the flow chart of data transferring between the Web browser, the Web server and the CGI program.

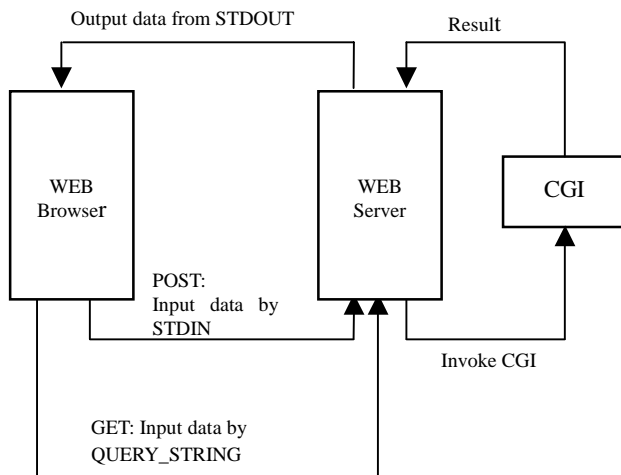


Figure 1. sketch map of CGI

As shown in figure 1, the Web browser sends the input data to the Web server. The Web server sets up environment variables based on the received data, then forks a new process to execute the CGI program. The CGI program reads the data from environment variables and outputs the results in HTML format to STDOUT. The browser displays the returned results to users. Here should emphasize that the CGI is a script program, which is independent to the web server and be triggered by web server.

3. THE CGI IN EMBEDDED WEB SERVERS

3.1 Design Theory

Generally speaking, embedded Web servers are applied in special fields. In most cases, the process flow of control devices is fixed. As a result, the Web server with any of the interactive function can be used in control devices. On the other hand, the standard CGI cannot be implemented in the 8-bit or 16-bit MCU system based on the analysis above. In this article, an approach to implement the Web server in 8-bit MCU system is proposed according to the problems discussed above. This approach has a remarkable practical significance in low-end devices' evolution because it meets the requirement of low-cost and high-usability. This Web server only supports the CGI method without providing any other function of the standard Web servers. Users can achieve basic interaction with control devices via this Web server. We move the independent file of standard CGI inside the Web server and take it as a function. Users can call it through the API when developing a specific application. The functions developed by the end users should be compiled with the Web server itself and run as a single program.

To design an embedded web server suitable to the 8-bit and 16-bit MCU systems in MCU system, some rules should be observed. First of all, it is based on the HTTP and CGI

standard, and is conform to the basic Web theory. Secondly, CGI is part of the whole Web server instead of an independent executable file. Third, the CGI program can read data directly from socket instead of STDIN and the CGI program can write the data directly to the socket instead of STDOUT. And finally, no environment variable is needed because the CGI program and the Web server are in the same program.

3.2 Procedure

The Web server receives and analyzes the requests from the client. If there is no data received from the client then the server calls `get_status()` to obtain current states of the device and display them to the browser. Users can set new parameters from the client form and submit them to the server. The server uses the CGI function to parse and process the received parameters. Then it uses the parameters to set the device by function `set_status()`. The states of the device are returned back to the client simultaneously. Thus, users can monitor remote devices by continuously interacting between the client and the server.

3.3 Embedded Web Server Flow

```

step0: begin
step1: read and separate request
step2: if not CGI, send default HTMLpage,
      goto step 8
step3: set CGI parameters
step4: if "get", read data from QUERY_STRING
      if "post", read request body
step5: analyze FORM data
step6: obtain and set status
step7: send encapsulated data to browser
step8: ready to next request
  
```

3.4 Main Functions and Their Descriptions

void do_service(int sd) ;

This is the main function. It reads and analyzes the requests and sends the received data to the CGI function, and transfers the device states or the modified states to the client in HTML format.

request_rec

****parse_request_line(request_rec *r) ;***

To analyze the request head and parse the received request head into three parts, i.e. request method, URL, HTTP version.

request_rec

****handle_filename(request_rec *r) ;***

To obtain the URL data after " ? " when the 'GET' method is used. These data are input by users and are encoded. It seems like a list name=value pairs connected by a ' & '.

header_rec *parse_header(char *array, header_rec *header, char c)

To parse the request head into name and value two parts respectively. For example, Content_length : 58 is parsed to: header -> name = Content_length and header->value = 58. It will be used later.

void cgi_process(char *input) ;

To parse the data submitted by the form. In fact, it is the function completed by the standard CGI program. Because this Web server cannot support the standard CGI, so it is moved inside the server. This enhancement is according to the CGI standard and can be implemented in the 8-bit MCU system at the same time.

void pack_html_formation

(char *form CGI[]) ;

To pack the result into HTML format and wait to send it to the client later. The function do_service(int sd) completes the sending work.

void send_status_line(char *array,int sd) ;

This function sends the status line to the client end. In this Web server, only a simplest fixed status line is needed. e.g. "HTTP/1.1 200 OK\r\n".

void send_white_line(int sd) ;

This function sends a blank line to denote the end of a response head.

3.5 User API

Embedded Web server software must provide mechanisms for the embedded application to generate and serve Web pages to the browser and to process HTML form data submitted by the browser. Designing the interface between the embedded Web server and the embedded application offers the greatest challenge. The user here is the programmer of specific control devices who use our embedded Web server. Because the CGI is an integrated part of the Web server, some functions should be offered by users to interact with the Web server. These functions are compiled with the Web server as a program then are programmed into a flash memory in MCU system. This MCU now becomes an embedded Web server to control all kinds of specific devices.

void get_status(char *stat_list[])

This function obtains current status of the device under control. It is completed by users according to the specific device. These data are translated into characters and put into the array of stat_list[] under a special order.

void set_status(char *stat_list[])

This function fetches the new parameters from stat_list[] and sets up the device with it. Thus, the user can control remote devices by submitting parameters with a browser.

The two functions above are written by users and can be invoked by the Web server through a simple interface. We can definitely design more accurate interface according to the architecture forenamed. Because the interfaces should be designed based on the specific device, we only give the simplest example for brevity.

void html_formation(char *form CGI[])

This function used by users is to design a suitable HTML format to be displayed in panel. The fixed data only should be put in the corresponding places in form CGI[], while the temporary parameters should be initialized by " " .

4. CONCLUSIONS

Embedded systems have limited CPU and memory resources available to them. In many cases, these limited resources are largely committed to mission critical and real-time applications. For example, a data communications router may require most of the CPU and memory resources of the embedded system to forward packets. Unless the embedded Web server software is sensitive to the real-time requirements of mission-critical applications, it might drop packets or lose data. It is most significant to design the embedded Web servers with low-cost, high-usability properties and suitable to 8-bit and 16-bit MCU systems. At this paper, the CGI designed in embedded Web server was addressed thoroughly. At the same while, the implementation was discussed in full details. This Web server has been already implemented in the

embedded systems with an 80C196 or 80C51 core.

Nowadays, embedded Web technology is the hotspot in computer field.. This technology will accelerate the progress to make low-end control devices become interconnected. When developing the Web program in embedded systems, we should consider the general Web technology as well because it is the standard we must abide. But the embedded system has its own specialty different from the general environment. The common Web technology without any change cannot be implemented in embedded systems. It should be modified to embedded systems according to the specification and the standard. Implementations of CGI in Embedded Web Server detailed in this paper have made a successful try.

5. REFERENCES

- [1] Kyoko Matsuura, Tadahiro Hara, Akashi Watanabe, Tatsuo Nakajima, "A New Architecture for Home Computing", IEEE WSTFES'03
- [2] Terry Hess, T.H.E.Solution LLC, Greer,SC, "Accessing Devices Using a Web Service", Proceedings IEEE SoutheastCon 2002
- [3] McCombie B, "Embedded Web servers now and in the future", Real-Time Magazine, March 1998
- [4] Wilson A, "The Challenge of embedded Internet", Electronic Product Design, January 1998
- [5] Ian Agranat, "Embedded Web servers in Network Devices", Communication Systems Design, March 1998
- [6] Cao jian, Li wen-bin, Zhang jian-feng, "Instance Tutorial of Perl5、PHP4 and CGI", Beijing, Publishing House of Electronics Industry, 2000.12, 7-39
- [7] Wang xiao-ping, Zhong jun, "Advanced Network Programming with Visual Basic", Beijing, People's Postage Publishing House, 2001.4, 162-273
- [8] RFC 2616, Hypertext Transfer Protocol --HTTP/1.1, June 1999
- [9] RFC 1867, Form-based File Upload in HTML, November 1995
- [10] <http://mcu21cn.topcool.net>
- [11] <http://www.chnibs.com>



Zhang xi-huang is the associate professor of the school of information engineering, Southern Yangtze University. Now, he is a PhD Candidate of Southern Yangtze University. His research interests are embedded systems and industrial

control.

Xu wen-bo is a professor and Ph.D supervisor of the school of information engineering, Southern Yangtze University.

Modeling Distributed Systems: Architecture and Process*

Beihong Jin Jianchao Wang
Technology Center of Software Engineering, Institute of Software,
Chinese Academy of Sciences, Beijing 100080, China
Email: jbh@otcaix.iscas.ac.cn Tel: +86(0)10-62567330

ABSTRACT

With the rapid prevalence of distributed applications, more attentions have to be focused on modeling distributed system. This paper investigates the characteristics of constructing a distributed system, then gives a series of design considerations of distributed software architecture including component interaction, security, failure handling and QoS, and presents the suggestive modeling process which extends the regular software process. Case study shows our suggestions on modeling are valuable for constructing distributed systems.

Keywords: Modeling, Distributed System, Software Process

Various distributed systems have been built up in recent years, which can be chiefly attributed to the development of network technology and the constantly decrease of hardware price. Distributed computing environment can provides better price/performance ratio, and the powerful concurrent computation ability as compared with the main frame. Another reason for the fast growth of distributed system is that many applications themselves are distributed. For instance, data processing is often distributed in the modern enterprises which usually comprise several companies or subsidiary-companies scattering in different locations. What's more, there are large amount of information exchange and analysis among these enterprises, as well as between enterprises and their partners or their customers. All these application requirements move forward the researches in the constructing of distributed systems.

In order to meet the requirements of the distributed system development, we must set up a whole set of software modeling methodologies, and draw out the process for constructing a distributed system. Modeling a software system is to describe the target system with an abstract language which is either formal or informal. Up to now, general modeling techniques and some specific ones for concurrent and real-time systems have been studied [1]. In this paper we focus on discussing the way of building up a distributed system from the view of modeling a software system.

This paper is structured as follows. Section 1 introduces the characteristics of building up a distributed system. Section 2 presents a series of key points on the software architecture of the distributed system. The suggestions of expanding the general modeling procedure are provided in section 3. After that we show a case study which applies the proposals offered in previous two sections. In the last section, we conclude the paper with a summary.

1. CHALLENGES OF CONSTRUCTING A DISTRIBUTED SYSTEM

Distributed system is a software system that incorporates several computing units through network interconnections to complete computation tasks. It provides the end users with a single computation environment to share resources and cooperate with each other.

In constructing a distributed system, several issues below should be solved [2, 3]:

- Heterogeneity: Distributed systems should be capable to be constructed on different networks, operating systems, hardware and with different programming languages.
- Transparency: Distributed systems should provide some degrees of distribution transparency, whose forms include access transparency, migration transparency, location transparency, relocation transparency, replication transparency, concurrency transparency, failure transparency and persistence transparency. The transparency supported by a system is tightly correlated with its application. In fact, a 100% transparency does not practice well in a realistic system because of the performance loss brought by excessive emphasis on the transparency.
- Openness : A distributed system should be open, and the service specification it provides should be complete and platform-independent.
- Scalability: a distributed system can be scalable with regard to its size, geography and management. From the view of system size, a distributed system is scalable if the system resources to be added to the distributed system are constants when a new user is added to the system. Higher scalability should be satisfied as the system is being designed.
- Security: There are various security attacks such as eavesdropping, masquerading, tampering, denial of service. So we should provide corresponding ways to ensure the privacy, integrity and availability for users to share resources.
- Failure handling: Any single process, computer or network of a distributed system has the possibilities to fail respectively, thus every component should have knowledge about the failures that may occur internal or external such as in its dependent components, and can deal with these failures properly.
- Requirements on the QoS(Quality of Service): The distributed system shows diverse usage styles, for instance, components of the system may experience different loads of work(e.g. some web pages can be hit several millions times a day), some part of the system

*This work was supported by the National Natural Science Foundation of China under Grant No. 60103008, the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312005.

may be disconnected or under an unsteady connecting situation (especially when the system involves mobile computers), and some applications (mainly the multi-media applications) have special requirements on the bandwidth and delays of the communication. Moreover, the scales of systems vary greatly according to the environment of the systems. All these features remind us to give special attention to the QoS of the distributed system when it is under construction.

2. SOFTWARE ARCHITECTURE OF DISTRIBUTED SYSTEM

As M. Shaw and D. Garland defined in [4], software architecture should include the description about the components which build up the system, the relationships among components, styles of the system composition and constraints to be satisfied.

We must identify the types of components in the system which we are about to set up at first. In terms of architecture style, if the users use the system in a request-response way, the system will be analyzed to employ the client-server architecture (including browser-server architecture), and correspondingly there will be client component and server component in it. Moreover, the system can be divided into more than one server component so that several servers may complete the system functions cooperatively if there is too much functionality in one single server. What's more, in order to reduce the cost on the hardware and software installation/support management, the thin-client policy is also applicable in which the client only deal with the window-based user interface while remote computation server provide functions such as querying and calculating. If components in the system cooperate with each other without a master/slave separation in processing a distributed activity or computation, their relationship is peer-to-peer. For example, a distributed write-board application, which allows users in different computers to browse and modify the shared canvas/document interactively, deploys one component in each place. This component is responsible for completing group communications and event notifications, thus enables all the users to be aware of modifications on the canvas/document in time.

Next we should make out how the components interact and coordinate with each other. Below outlines the interaction-coordination requirements existing among distributed components:

- Describing method of the components interacting information: Components interact and coordinate with each other by the means of information passing, therefore, there are several things for the message needed to be confirmed, such as whether to hide the differences of hardware and OS in message passing, whether to be capable to describe message's application semantic by itself – that is, whether we can recognize the message's structure and the data types it contains. By applying proper external data representation such as CORBA CDR to encode-decode the data which are transferred among components, the differences of hardware and OS under components can be hidden.
- Degree of referencing coherence among components: A high restriction of the referencing coherence means that

there are requirements about the component transferring. The component transfer mode can be further classified. For example, is the transferring initiated by the sender or the receptor? Is the transferred component executed by the component which takes charge of the transferring, or started to execute independently? The widely used example of component transferring is applet, which enables the downloading of codes on Web server and executing of codes on the client. Here it is the receptor that claims for code to be transferred to its location and run mobile codes at the browser's address space. Currently, many applications adopt the mobile agents, which can autonomously transfer among heterogeneous/homogeneous computers for performing a task.

- The degree of timeline inherence among the components: That is, whether are there any restrictions on the time of the executions or interactions of components? The synchronous communicating modes such as RPC, RMI can be used on the systems with the confine of interactive time, while the asynchronous modes such as message queue communication can be used in systems without such restrictions.
- The degree of timeline coupling among components: It judges whether there is necessity for two communicating processes start and run at same time. Lower timeline coupling means the component can be executed relatively independently.
- The degree of reference coupling among components: This is to say if there is necessity for the two parties of the coordination to know each other. For instance, components may interact with each other through pub/sub interaction or shared storage mechanism without knowledge of their counterparts [5]. The former is usually called meeting-oriented coordination, while the latter generative communication coordination.

Apart from identifying the components and the interactions among the components, we need to take into a total consideration the failure-handling policy, security policy and methods to ensure the QoS of the system.

A lot of component communicating failures may be happened in the distributed system such as channel omission failures, send/receive omission failures. Component failure such as component crash, component arbitrary failure may also be occurred in the system. Commonly the redundancy techniques, including information redundancy, time redundancy, physical redundancy, are used to mask above failures except arbitrary failure.

Concretely, in order to mask the component communicating failures, reliable communication mechanisms are adoptable, for example the RPC with At-Most-Once semantic, or the RPC with At-Least-Once semantic if the operations on the components are idempotent. For the crash failure of the components, there are also several methods introduced: We can specify the policy of replacing component process, which means to resume the system by replacing the component process and then leverage the information stored on the hard disk. And we can also choose component replication policy to replicate components and disperse them onto several locations so that even if one component or processor has faults other

components can provide correct service in time. In order to avoid the possible component crashes in the distributed system, data replication method is often adopted which keeps multi-copies for a file so that the component can turn to other copies when one of them invalidates. Moreover, when facing a group of operations within a component, the atomic transaction is also adoptable in order to ensure the ACID feature of this group of operations. Regarding dealing with failures, it is also a common way to convert the failures into more acceptable ones, for instance, checksums are used to mask corrupted messages, effectively converting an arbitrary failure into an omission failure.

There are mainly two types of security threatens in the distributed system: unauthorized object access and threatens to components (including mobile components) and communication channels. Lacking trustful info about the message source is a primary factor influencing the security and performance of the components, which means that under a circumstance without security mechanism, the request-receiving components can not decide the identity of the request sender, and the response-receiving components can not tell whether the response came from an expectant component or a malicious pretender. Besides, when the information is transmitted through the web, it is possible to be intercepted, modified, forged and replayed. In the meantime, system services may be made excessive futile invocations, leading to deny access for normal requests because of overload of the physical resources such as network bandwidth and process ability of the server. In order to cut off these potential security threatens, we need to establish a set of security policies and mechanisms:

- Establishing authentication mechanisms, including identity authentication and data source authentication to ensure that a pair of communication process can verify its counterpart's identity, and the message source, message create time and data transferred through the channel can be decided. This is mainly achieved with the aid of cryptology technologies.
- Building up authorization and access control mechanism on the network level, operating system level, database level and application level. Based on the application requirements, we can adopt different access control mechanisms such as Discretionary Access Control, Mandatory Access Control, Rule-Based Access Control, etc.
- Setting up audit mechanism which makes it easier to analyze and trace by recording the users' operations in the system.

Cryptology plays a critical role in ensuring the system's security, for example, identity authentication can be implemented by using symmetrical encryption techniques or asymmetrical encryption techniques (Public Key Encryption Technique), and the data-source authentication can be implemented through secure channel and MAC (Message Authentication Code). Moreover, we can set up a general 3A (Authentication, Authorization and Audit) mechanism in the actual application.

Besides basic system functions, most distributed systems have a demand for QoS (Qualities of Service) which derives from goals and end-users of the system. For general distributed

systems, the QoS includes availability, reliability, performance, maintainability. QoS vary greatly in different types of systems. For example, in transaction process system, the transaction throughput is an important performance factor, while in multimedia system, the network bandwidth, latency, jitter and package loss ratio are of chief concern.

Below are some common ways to provide QoS management in distributed systems:

- Integration: This method is to modifying the system kernel directly, tightly coupling QoS with the system. It seems to provide better performance due to the possibility to optimizing the system on diversified facets.
- Implementation as particular service: This method does well in portability and interoperability; however, it has no guarantee on strict end-to-end time constraints and shows potential security leaks.
- Implementation by intercepting: We can intercept the messages on the components communications channels or to intercept the system processing flow to plug QoS codes, using Interceptor, Reflection or other patterns.

Research work done has diversified the ways to perform QoS tasks. For example, in order to enhance the system performance, we can apply load balance which can be implemented not only by searching solution space but also by mathematic methods such as linear programming, or anneal simulation originated from thermodynamics, or genetic algorithm from biology. In addition, the general techniques should vary with different system requirements. For example, data replication and cache technique are widely used methods to enforce the system reliability and enhance the performance. However when they are applied in the actual environment, we must firstly give the definition of the consistency of the replicated data which is needed in the current application (The most frequently used data consistency definitions include lineal consistency, sequential consistency, weak consistency and so on), then we can realize the consistency protocol which defines how to updated these data copies basing on one of above definitions.

Nowadays, the distributed system not only must satisfy the corresponding QoS requirements, but also should have the abilities to negotiate, customize and adjust the QoS dynamically. For instance, it is often respected to decide the reliability configuration of the system basing on both the failure report of the system and the expectation of the users.

3. MODELING PROCESS OF DISTRIBUTED SYSTEM

Modeling a distributed system should follow the fundamental framework of the software modeling process. For simplicity, we only concentrate on the modeling in the requirement and system designing. We will extend the general analyzing and designing process [6] to form a constructing process which is suitable to build distributed systems [7].

During the requirement analyzing phase, it is needed to capture the requirements on the component interaction, failure handling, system's security and QoS of the system. Typically the requirement analysis turns out to be a set of system models

which we call requirement specifications. The requirement specifications can be represented either formally such as by CCS (Calculus of Communicating Systems) and PetriNet, or informally such as by data-flow diagram, or use-case diagram, class diagram and state diagram in UML. Requirement specifications record the functionality and non-functionality requirement. Besides these, physical architectural model is also needed. This model, usually called hardware platform specifications and represented by UML's deploy diagram, describes the hardware platform deployment of the system.

Below we present the basic steps of the object-oriented requirement analysis. We use the UML for describing the results of requirement analysis:

1. Identify the scenarios or use cases to acquire the users' requirements. Identify the communicating mode, potential failures, security requirement and QoS requirement of the system, and describe them with UML's use-case diagram, deployment diagram.
2. Define the classes and objects basing on requirements, and specify their attributes and operations, and describe them with UML's class diagram.
3. Define the hierarchy of the classes: define the inheritance and composition relationships among classes to form sub-systems, and then describe them with UML class diagram.
4. Build up Object-Relationship model which consists of the static parts of the system, and describe it with UML class diagram.
5. Build up Object-Behavior model which defines the dynamic parts of the system, describe the sequences and events that result in state transition, and describe it with UML state diagram, sequence diagram and collaboration diagram.
6. Review the analysis model with use cases or scenarios.

The interaction model of the distributed system has an effect on the design of system architecture, especially that of the components, while the failure-handling and system security will infect the system's strategy to handle exceptions, and what's more, the requirements of QoS towards the design of the system are more extensive but there is no effective way to represent them at one particular step currently.

Below are basic steps of the Object-Oriented design. Still, deliverables are described with UML.

1. Design of the System

- 1.1 Divide the analysis model into sub-systems
- 1.2 Categorize the system's distribution and concurrency. Make sure whether the system is distributed by examining the deploy diagram. Determine the concurrent tasks by examining the state diagram and concurrent users' status.
- 1.3 Design the architecture of the system and dispatch subsystems to the processors/ tasks Determine how the subsystems and objects are distributed on the network nodes, and decide the communication model among subsystems such as client-server style, or peer-to-peer style. Make sure what kinds of communication failures may exist in the system and their corresponding handling methods, and then determine the policies of

communication security. More concretely, we can deploy each subsystem to an isolated processor, or deploy all the subsystems to one processor in which the operating system is responsible for managing the concurrent events. Thus we can determine the system runs on multiple processes or multiple threads. Furthermore we must give the concurrency control policies, the task scheduling policies, the ways in which tasks are activated, load balance strategies, and the task scheduling policies as well.

- 1.4 Mark global shared resources and their access control mechanisms. It's necessary to take into consideration whether data replication is needed, as well as the policy used for data consistency.
- 1.5 Choose design for implementing data management.
- 1.6 Design user interface.
- 1.7 Consider how to handle exceptions, including how to handle failures, what about the system's ability to tolerance failures is, and what kinds of security facilities the system should have in the face of security attacks.
2. Design of objects
 - 2.1 Present design descriptions of each object. Design every operation including the its algorithm and its data structures at procedure level.
 - 2.2 Define internal classes as well as internal data structures in classes.
 - 2.3 Design messages that are exchanged among the classes.
3. Consider all tradeoffs, review design model and iterate the design process if necessary.

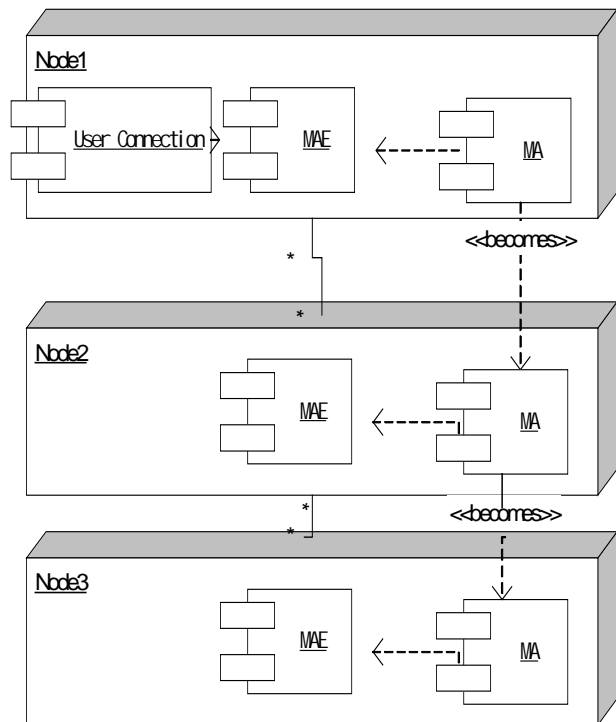


Figure 1 System Deployment

4. CASE STUDY

In this section we will illustrate the design issues and modeling process of the distributed system with an example: *Budget*

Travel Guiding System

While the user inputs travel requirements such as destination, duration, traffic and accommodations requirements, the system will automatically arrange the travelling schedule and inform the user. The *Budget Travel Guiding System* is required to support mobile user, as well as auto-searching for servers which can provide related information.

Because the *Budget Travel Guiding System* is an interactive system, it has restrictions on the response time, which determines its high coupling and cohesion on timeliness. Besides, the system is required to provide ability to automatically search related servers, thus there should be loose coupled references among the components. Based on these concerns, the system will be difficult to extend if it adopts the client/server structure, and the clients' frequent accesses to the server will turn it to the bottleneck. On the other hand, functions such as information searching, travel scheduling are relatively isolated, so if we use mobile agent mode on the information searching component, there will be better scalability for the system which make it easier to add servers to the system. To sum up, we come to the conclusion that a mobile agent system which uses P2P structure will be best fit for above requirements. That is to say, we believe that enhancing the degree of reference cohesion among the components will provide better QoS for the system.

Next we will give some main steps in the modeling of the *Budget Travel Guiding System* according to the modeling process discussed in the Section 3.

- Requirement Analysis Step 1: Identify the scenarios or use cases to acquire the users' requirements.

System requirements include: request submission, auto querying, travel agenda arrangement, response acquiring and displaying. The user connects local Mobile Agent Environment (MAE) in which the mobile agents run, and the system delivers the work such as information querying to the mobile agent. The mobile agent can autonomously transfer to and run in several MAEs to complete related information query. The Mobile Agent Environment should be deployed to all possible locations where the application may be engaged. Assuming that the system is deployed to 3 nodes, figure 1 gives its deploy diagram. There are two kinds of nodes; one has user connection while the other has not. Nodes communicate with P2P mode. The "becomes" link represent the transition of mobile agent between MAEs. Obviously we can see the system is distributed.

Possible failures in the system include:

- ◆ Omission failures: Because the system is required to support mobile environment, it may lose the transferring messages.
- ◆ Crash failures: Nodes and components at nodes such as MAE all have possibilities to crash.

Requirements on system security include:

- ◆ Authentication to the users' identity.
- ◆ Authentication between MAEs, and MAE's authentication to the mobile agents arrived.
- ◆ Authorization to mobile agents when they access local resources.

QoS requirements include:

- ◆ Availability: MAE must ensure that it can respond to user requests timely and the users can receive the responses from MAE in time.
- ◆ System performance: It includes the system's throughput to handle users' requests, the response time to users' request, and the degree that this response time matches the user's expectation.

- Requirement Analysis Step 4 : From above requirements we can identify several base classes in the system, for example the MessageDispatcher class which handles messages for user connection, MAEcontext class in the MAE which records runtime information about the MAE, and MAgent class which processes concrete scheduling task at runtime. Then we design these classes in detail and derive other classes from them if necessary. At last we describe these classes with UML class diagram.

- Design Step 1: During this step we find a series of sub-systems, including:

User connection subsystem:

- ◆ Message handling component: it marshals the user requests into data stream which is understandable for the mobile agent, and unmarshals the results from mobile agent and display it on the user interface.
- ◆ Communication component: it is responsible for tasks such as creating connection with local MAE, sending and receiving messages.

MAE subsystem:

- ◆ Communication component: It handles the communications between MAEs, including secure transferring of mobile agents and message exchange among mobile agents, as well as between mobile agent and context objects.
- ◆ Security control component: It provides authentications among mobile agents, as well as between mobile agent and context objects, including examination on the validity of the mobile agents arrived.
- ◆ Resource access control component: It checks the validity of the mobile agent's accesses to local resources according to its access permission.
- ◆ Mobile agent runtime context: It is responsible for the managing and scheduling of the mobile agents, including their creation, suspension, resume, transferring and cancellation. It is also in charge of storing related states and handling failures in the process of the system. For instance, it should handle lost mobile agent resulted from the network disconnection or host's shutdown, as well as dead agents which are caused by system crashes.

- Design Step 2: Construct object-relationship model and object-behavior model basing on the components and the class relationship in them. Then draw corresponding UML state diagram, sequence diagram and collaboration diagram.

After the analysis and design mentioned above, we can establish the framework of the *Budget Traveling System* and master the key points of the system development.

5. CONCLUSIONS

The distributed system is in a rapid developing period in recent years, thus its modeling method is also under great attention. In this paper we first explained challenges in constructing a distributed system. Then, aiming at the construction process, we presented design issues on the software architecture including identifying components and their relationship, and gave some common problems and their corresponding solutions regarding security, failure handling and QoS of the distributed system. In order to meet the requirements of modeling a distributed system, this paper also provide a suggestion extending the general modeling process. At last, a distributed system case *Budget Travel Guiding System* is studied to illustrate our analysis strategy and the effects of the modeling process in constructing a distributed system.

6. REFERENCES

- [1] Selic B, Rumbaugh J. *Using UML for Modeling Complex Real-time System*, White Paper, Rational Software Corporation, <http://www.rational.com/media/whitepapers/umlrt.pdf> 1998
- [2] George Coulouris et al., *Distributed Systems Concepts and Design* (3rd ed.), Addison-Wesley, 2000.
- [3] A. Tanenbaum, U. Van Steen, *Distributed Systems: Principles Paradigms*. Upper Saddle River, NJ: Prentice Hall, 2002.
- [4] M. Shaw, D. Garlan, *Software Architecture: Perspectives on an Emerging Discipline*, Prentice Hall, April 1996.
- [5] G. Cabri, L. Leonardi, F. Zambonelli, *Mobile-Agent Coordination Models for Internet Applications*, IEEE Computer, February 2000
- [6] Roger S. Pressman, *Software Engineering, A Practitioner's Approach* (5th ed.), McGraw-Hill, 2001
- [7] Fethi A. Rabhi and Sergei Gorlatch (Eds), *Patterns and Skeletons for Parallel and Distributed Computing*, Springer-Verlag London Limited 2003



Beihong Jin is an associate professor and deputy director of Technology Center of Software Engineering, Institute of Software, Chinese Academic of Science. She received her B.S. degree in Computer Science from TsingHua University in 1989. She received her M.E. and Ph.D. degrees in Computer Science from Institute of Software, Chinese Academic of Science in 1992, 1999, respectively. Her research interests focus on distributed computing concerning transaction processing, service-oriented system, and software engineering issues

Opening Component Integration Architecture*

Zhou Xiaofeng¹⁺, Wang Zhijian¹, Fei Yukui^{1,2}

¹ School of Computer & Information Engineering, Hohai University, Nanjing 210098, China

² School of Information Engineering, Shandong Agriculture University, Taian 271018, China

⁺ Tel: 86-025-83786520, Fax: 86-025-83735625, E-mail: z_xiaofeng@sina.com

ABSTRACT

The software component technology is one of primary technology to improve degree of software reuse, thereby to solve crisis of software producing. What realizes flexible believable software reuse has become imperious need of software producing in open dynamic environment as Internet, along with applications mostly base on Internet. It is primary way of software component development to expediently realize issuing and reuse of software component situated on every nodes using Internet, accordingly to form software web similarly of now information web. This paper advances an opening component integration architecture (OCIA) to fulfill heterogeneous component integration under environment of Internet using web services technology, and present realization steps based on OCIA.

Keywords: OCIA, Web services, software web, component integration

1. SUMMARY

The concept of software component comes of NATO's software engineering conference in 1968. McIlroy advanced concept of software component, component factory in paper of "Mass-produced software Components"[1]. In 1970s and 1980s, the software component primary points reusable program code segment, commonly called code-ware. In 1990s content of software component was extended, primary include analyze-ware, design-ware, code-ware, test-ware, etc. Because software reuse has variety we can divided it into production reuse and process reuse [2]. The software component provides feasible resolve project for software reuse and solving software produce crisis.

In latest years, the cognition to the component has occurred new change along with distributed object, Internet, JAVA, Client/Server, etc technology development. Most people consider that software component is software unit solely developed and having particular function, it is used to integrate application system with other components.

Component based development (CBD) is software development method using commercial off-the-shelf to integrate software application system according application needs [3][4]. The primary content of CBD research include: component obtained component model, component description language, component class and searches, component integration, standardization, etc [5]. The component integration is core of CBD, final purpose of other research contents is for easily realizing component integration.

Now the component integration is divided generally into black-box integration model, white-box integration model and gray-box integration model. The divisory gist is degree to need understand component inner detail when component is integrated. The methods of component integration have primary framework based method, connector based method and glue code based method.

But current component integration methods primary aim at stand-alone environment, namely all integration must finish in local. First step must obtain component and the component obtained can run at local environment, otherwise the component obtained will can't be integrated. These component integration methods must know needs of component running, so it is difficult to realize integration of different type component using them.

What realizes flexible believable software reuse have become imperious need of software producing in open dynamic environment as Internet, along with applications mostly base on Internet. It is primary way of software component development to expediently realize issuing and reuse of software component situated on every nodes using Internet, accordingly to form software web similarly of now information web. This paper advances an opening component integration architecture (OCIA) to fulfill heterogeneous component integration under environment of Internet using web services technology, and present realization steps based on OCIA.

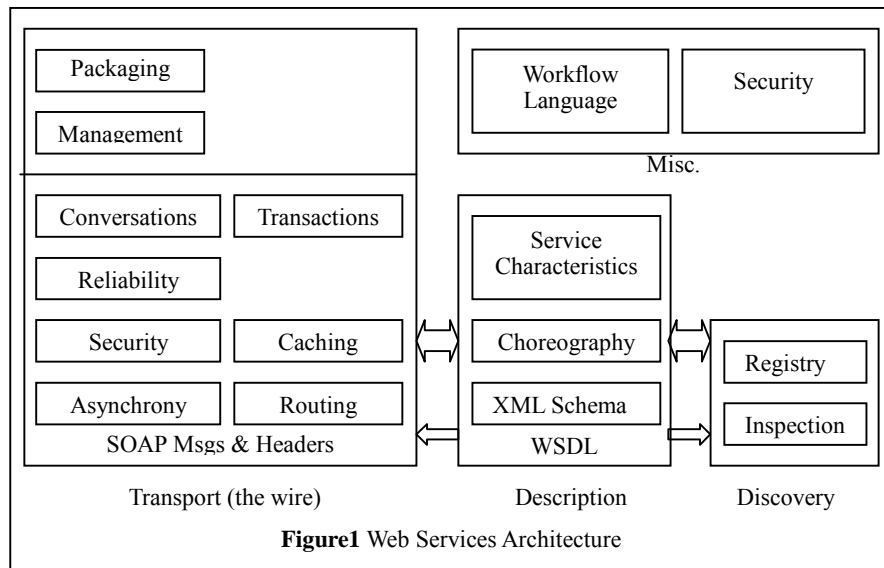
2. OPENING COMPONENT INTEGRATION ARCHITECTURE

2.1 Web Service Technology

Web service is an extension of object/component technology in Internet, it is an object/component technology installed on Web. The web service has both stand-out capability of developing module based components and web. First, the web service has the function of black-box linking component, can be reused under the circumstance not caring for how the function is achieved; At the same time, the web service differing with traditional component technology, provides cooperation between them through integrate different types function modules opened out different platform. So web service is defined as a model of next generation distributed systems development used widely.

Now, the web service technology has given broad self-identity. W3C is instituting correlative standards. Documents produced by W3C include SOAP version 1.2 and WSDL Version 1.2 etc. Web services Architecture consist of 3 stacks that are closely related. A transport stack is for standards that are exchanged on the wire. Description is for describing an individual or collection of services. Discovery is the finding of services (show like figure 1).

*Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA113170; the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312002)



This shows that the protocols related the web service include Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), Universal Discovery Description and Integration (UDDI). Many of higher layer protocol wait for opened up, such as routing, reliability and transaction protocol.

SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using an XML document. SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and mechanisms for encoding application defined data. This allows SOAP to be used for a large variety of purposes ranging from messaging systems to remote procedure call (RPC) invocations [6].

WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate [7].

Universal Discovery Description and Integration is a specification for distributed web-based information registries of Web Services. UDDI is also a publicly accessible set of implementations of the specification that allow businesses to register information about the Web Services they offer so that other businesses can find them. The core component of the UDDI project is the UDDI business registration, an XML file used to describe a business entity and its Web Services. Conceptually, the information provided in a UDDI business registration consists of three components: “white pages” including address, contact, and known identifiers; “yellow pages” including industrial categorizations based on standard taxonomies; and “green pages”, the technical information about services that are exposed by the business. Green pages

include references to specifications for Web Services, as well as support for pointers to various file and URL based discovery mechanisms if required [8].

The Web Services Flow Language (WSFL) is an XML language for the description of Web Services compositions. WSFL considers two types of Web Services compositions. The first type specifies the appropriate usage pattern of a collection of Web Services, in such a way that the resulting composition describes how to achieve a particular business goal, typically, the result is a description of a business process. The second type specifies the interaction pattern of a collection of Web Services; in this case, the result is a description of the overall partner interactions [9].

The Web Services is a good protocol when it is used to provide information service. But it is not enough to provide other services such as software services, grid services. Of course the Web Services has good expansibility, so it can easily suit different needs through extended. This is why we design OCIA using Web Services technology.

2.2 Opening Component integration Architecture

The opening component integration architecture is a software component integration architecture designed to be used environment of Internet and aim at characteristic of software component based on Web Services. The opening component integration architecture includes primary three parts: component description, component discovery and component integration (show like figure 2).

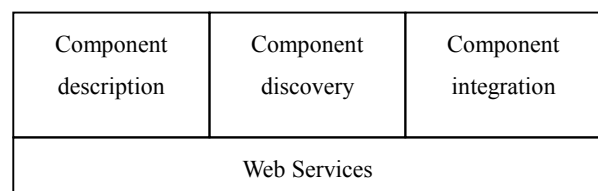


Figure 2 Opening Component Integration Architecture

Three parts are closely related. The component description describe software component using extended WSDL for discovery and use of software component. The component discovery is used to issue and find component on Internet. The component integration integrates the application system using

software component found.

2.2.1 Component Description

Because software component is a sightless black-box its description is a only route what user know and use component, and emerge special component description language such as π , CDL, CIDER, LILEANNA, RESOLVE and OOMIL etc. Recently the research using XML to describe component has occur because XML is degrees maturation and is abroad used. Some production has published such as IBM's Bean Markup Language (BML), xADL of C2 system of University of California, Irvine, etc.

Using component description language according with Internet standard is necessary for realizing software component integration at environment of Internet. Although using XML to describe component can reach this aim, but it is not enough convenience and it is only a standard of information exchange, so using extended WSDL to describe software component is better choice in the OCIA.

WSDL is one of standard of W3C, uses XML document, has itself standard description component such as Type, Message, PortType, ServiceType, Binding, Service. Its use is more convenience. But standard WSDL is defined for web services description, it can't suit completely needs of software component description, so it is necessary to extend it.

There are two aspects extension to WSDL in the OCIA. First, the description of web service is primary description of interface, but description of software component includes behavioural, coordination, quality, terminology, task and marketing except interface. Second, web services are generally permanent service, but software components provide primary temporary services because if a lot of components use as permanent services it will waste vast resources and affect efficiency of shared. So current state information of component must be recorded using WSDL.

2.2.2 Component Discovery

The discovery of software component can be realized basically by UDDI because software component is described by extended WSDL. But the discovery of software component more pays attention to select most suited component from a great of useable components compare with discovery of web service, so it is more complicate then discovery of web service.

To discover most suited component it is necessary to extend UDDI. There are a great of matching arithmetic now. It is key how band together matching arithmetic with UDDI.

2.2.3 Component Integration

At present the methods of component integration have primary framework based method, connector based method and glue code based method. The framework-based method realizes component integration via adding component into predefined framework. The connector-based method connects different component and realizes integration of several component via devising connector. The basic start of glue code based method is to solve local no-match of component when component is integrated example discord of message format. Its essence is a kind of connector too. Itself can't be reused easily because it often represents code at given environment. All of these methods are method based on local integration, so they can't suit needs of component integration at environment of

Internet.

In OCIA we use method similarly WFDL to realize component integration, namely realize component integration using binding method based on devised workflow.

Of course the component integration is different with WFDL. There are primary two different, one is component tailored, and another is re-banded. The component obtained can't suit complete actual needs generally. So it is necessary to tailor component obtained for satisfied actual needs at component integration. The component is dynamic but the application integrated by component is permanent. So component band must have self-adaptability. When the component banded can't use the system should band automatically new usable component.

3. COMPONENT INTEGRATION BASED OCIA

The component integration has 4 step generally based OCIA: discovery and matching, tailoring, integration, re-band.

The discovery and matching of component find needed component from Internet using UDDI, and discover most matching component from plentiful discoverable component using suited matching arithmetic.

The tailoring of component tailors discoverable component based on actual needs of user. The tailoring of component is different with tailoring of generic application system. The tailoring of component is materially what screens needless functions of component because the component is invisible and unchangeable.

The component integration is process of banding component selected. The component integration of OCIA is different traditional component integration, it need not encase component into application system really, it only band needed component. This method has infinite flexible, can realizes integration of heterogeneous component.

The re-band of component is important step to enhance usability of application system. The application system will re-discover usability component and band automatically new usability component using re-band when a component banded application system can't use.

4. NEXT RESEARCH CONTENT

Currently the OCIA is only architecture of component integration. There are many programs needed research, such reliability of component, integration tools, etc. Next step of research will focus on practicability of OCIA.

The reliability of component is a key problem of practicability. There are a great deal components in Internet, how ensure the component discovered is needed and how ensure it is reliability. Are the showing functions of component as actual? Can these functions all be used? There is not way to ensure reliability of component.

It is necessary designing and realizing a suit of tools of component integration for OCIA. The tools will easily use and have self-contained function. It can support each steps of component integration.

5. CONCLUSION

The software reuse is primary means to solve software produce crisis. The software component is one of important technology of software reuse. At new environment of Internet realizing software component is key of software component development. Realizing software component integration using Web Services technology not only accords with direction of technology development but also realizes easily.

This paper provides only architecture of component integration; there are many problems to solve.

6. REFERENCES

- [1] McIlroy M D. , Mass-Produced Software Components, Software Engineering Concepts and Techniques. In: 1968 NATO Conference on Software Engineering, Van Nostrand Reinhold, 1976, pp.88-98.
- [2] Ruben Prieto-Diaz, "Status Report: Software Reusability", IEEE Software, Vol. 10, No. 3, May 1993, pp. 61-66
- [3] Xia Cai, Michael R. Lyu, Kam-Fai Wong and Roy Ko, Component-Based Software Engineering: Technologies, Development Frameworks, and Quality Assurance Schemes, Proceedings of the Seventh Asia-Pacific Software Engineering Conference (APSEC.00), 2000, pp: 372-379.
- [4] A.W.Brown, K.C.Wallnau, The current state of CBSE, IEEE Software, 1998, Sept., pp: 37-46.
- [5] Yang Fuqing, software reuse and relate technology, computer science, 1999, VOL. 26, NO. 5: 1-4
- [6] SOAP Version 1.2 Part 1: Messaging Framework, <http://www.w3.org/2000/xp/Group/>
- [7] Web Services Description Language (WSDL) 1.1, <http://www.w3.org/TR/2001/NOTE-wsdl-20010315>
- [8] UDDI Technical White Paper, http://www.uddi-china.org/pubs/UDDI_Technical_White_Paper.pdf
- [9] Web Services Flow Language (WSFL 1.0), <http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>

Zhou Xiaofeng is a senior Engineer in School of Computer & Information Engineering, Hohai University. He graduated from Hohai University in 1986. His research interests are in distributed computing and software component. He joined the National High-Tech Research and Development Plan of China and the National Grand Fundamental Research 973 Program of China.

Packet Transfer Delay of the RPR Rings in Comparison between the Store-and-Forward and Cut-Through Architecture

Yi Yang, Mingcui Cao, Ping Huang

State Key Lab. of Laser Technology, Huazhong University of Science & Technology,
Wuhan, Hubei 430074, P. R. China

Email: kimbery@163.com Tel.: 027-87557064

ABSTRACT

In this paper, the high and low priority packet transfer delay of the N nodes Resilient Packet Rings (RPR) in the store-and-forward architecture is analyzed based on the queuing theory. The result indicates that both high priority and low priority packets' delay increase with the node number N of the RPR rings. The high priority traffic has less packet delay than the low priority traffic at the same node number N. The increase of the low priority transfer delay is much larger than the high priority traffic with the increase of the node number.

Keywords: RPR , Packet Transfer Delay, Store-and-Forward , Queuing Theory.

1. INTRODUCTION

As a rising MAN (Metropolitan Area Network) technology, Resilient Packet Ring (RPR) is a new network technology base on the packet-switched MAN. Because earlier SDH and ATM technology couldn't support the burst IP data flow very well, many manufacturers bring forward the IP over SDH and IP over ATM projects. But these projects have many disadvantages, such as static state bandwidth distribution is inadequate and the bandwidth utilization is insufficient. Ethernet network technology can support IP traffic very well, but it lacks QoS, network recovery and protection and network management mechanism. So RPR technology emerges as the times require. Just like SDH/SONET, it has self-recovery capacity, QoS and supports the data packet very well the same as Ethernet switched technology^[1].

RPR achieves very high bandwidth efficiencies using a combination of techniques: such as spatial reuse, ring-level aggregation, bandwidth penalty for protection and QoS^[2]. Now IEEE802.17 RPR workgroup is discussing the type of the node's buffer schedule. From Fig.1, we can see the RPR system can send and receive packets in both ring directions through independent access the node units. The client of each node can be separated into two independent subunits, one for each direction.

RPR has two means to transmit data: store-and-forward and cut-through. In this paper, we will emphasize the previous means.

From Fig.2 we can see, for the model of store-and-forward node, RPR MAC can transmit data packets from four possible queues:

- 1) High priority packets from the high priority transit buffer.
- 2) Low priority packets from the low priority transit buffer
- 3) High priority packets from the client Tx high priority FIFO.

- 4) Low priority packets from the client Tx low priority FIFO.

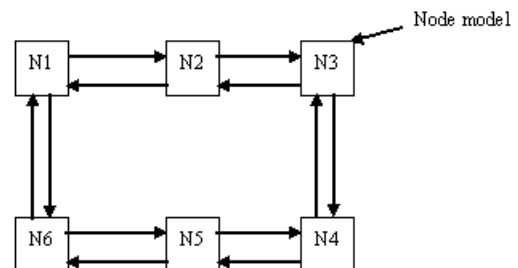


Fig.1 The model of the RPR rings with six nodes.

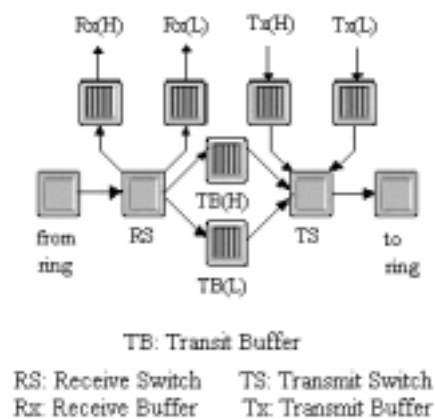


Fig.2 The model of store-and-forward node

The node sends the ring with its attached client traffic and forwards the traffic from upstream nodes to downstream nodes.

Transit frames are sent to the transit buffer and transmit frames are sent to the transmit buffer. The Transmit Switch (TS) schedules between transit frames and transmit frames. The TS is a strict priority scheduler where the transit frames have the highest priority.

In this paper, part2 established node models and analyzed the delay. The third part calculated the maximum network throughput in the store-and-forward architecture. At last we summarized this paper.

2. DELAY ANALYSIS

Any of the two counter-rotating rings is independent and can be regarded as the insert buffering ring. So we just discuss the single ring in RPR. In case of the broadcasting and control

data only occupying a small part of the bandwidth, we could neglect this part of data.

The average transmission delay is the time from which packet was stored in the Transmit buffer (Tx) to which was received by receive buffer (Rx). The delay in the Rx is ignored.

Now we use the store-and-forward approach to descript.

On the assumption that ^{[3][4]}:

- (1) The delay of one node is a constant.
- (2) The capacity of all buffers is infinity.
- (3) Waiting time and service time is independent.

The following assumptions are applied to high and low priority traffics independently:

(1) The arrival processes to each Tx(Hi), and Tx(Low), are Poisson distribution, with equal mean arrival rate $\lambda_{TxHi/Low}$ for all nodes.

(2) All nodes have the same distribution of the packet lengths, with associated packet sending times having first moment $\bar{X}_{Hi/Low} = 1/\mu$ and second moment $\bar{X}^2_{Hi/Low} = 2/\mu^2$, respectively.

(3) For N nodes RPR, all nodes have the same transmission pattern: node i transmits to node $[(i+j) \bmod N]$ with probability $q_{j,Hi/Low}$ ($j=1,2,\dots,N-1$) by the shortest path. Assume that the input processes of the buffer TB(Hi) and TB(Low), are Poisson distribution, with the mean arrival rate $\lambda_{TB, Hi}$ and $\lambda_{Tx, Low}$, respectively. The node model can be regarded as M/G/1 queuing system.

The formula for the waiting time of a packet with priority $k=1,\dots,K$ in a priority queue ($k=1$ highest priority) is :

$$T_{wk} = \left(1 - \sum_{i=1}^{k-1} \rho_i \right) \left(1 - \sum_{i=1}^k \rho_i \right)^{-1} T_r \quad (1)$$

With the mean residual service time of

$$T_r = \frac{1}{2} \sum_{i=1}^K \lambda_i \bar{X}^2_i \quad (2)$$

the average number of $_{Hi/Low}$ of TBs a packet traverses is:

$$\alpha_{Hi/Low} = \sum_{j=2}^{N-1} (j-1) q_{j, Hi/Low} \quad (3)$$

The relationship between TB traffic and Tx traffic in store-and-forward means is:

$$\lambda_{TB, Hi/Low} = \alpha_{Hi/Low} \lambda_{Tx, Hi/Low} \quad (4)$$

High and low priority traffics correlative affect network throughput of the insert buffer ring $_{tot} = _{Hi} + _{Low}$. We must notice that $_{tot}$ is just calculate one(unilateralism) ring. So the throughput of each high/low priority node is:

$$\frac{\rho_{Hi/Low}}{N} = \lambda_{Hi/Low} \bar{X}_{Hi/Low} \quad (5)$$

Now calculate the mean transmission delay:

In Fig.2, we define the priority $K=4$, the throughput of each

priority traffic is:

$$\rho_1 = \rho_{TB, Hi} = \alpha_{Hi} \frac{\rho_{Hi}}{N} \quad (6)$$

$$\rho_2 = \rho_{Tx, Hi} = \frac{\rho_{Hi}}{N} \quad (7)$$

$$\rho_3 = \rho_{Tx, Low} = \frac{\rho_{Low}}{N} \quad (8)$$

$$\rho_4 = \rho_{TB, Low} = \alpha_{Low} \frac{\rho_{Low}}{N} \quad (9)$$

With formula (1) we obtain:

$$T_{TB, Hi} = \frac{T_r}{1 - \left(\alpha_{Hi} \frac{\rho_{Hi}}{N} \right)} \quad (10)$$

$$T_{Tx, Hi} = \frac{T_r}{1 - \left(\alpha_{Hi} \frac{\rho_{Hi}}{N} \right) \left(1 - \left(1 + \alpha_{Hi} \right) \frac{\rho_{Hi}}{N} \right)} \quad (11)$$

$$T_{Tx, Hi} = \frac{T_r}{1 - \left((1 + \alpha_{Hi}) \frac{\rho_{Hi}}{N} \right) \left(1 - \left((1 + \alpha_{Hi}) \frac{\rho_{Hi}}{N} + \frac{\rho_{Low}}{N} \right) \right)} \quad (12)$$

$$T_{TB, Low} = \frac{T_r}{1 - \left((1 + \alpha_{Hi}) \frac{\rho_{Hi}}{N} + \alpha_{Low} \frac{\rho_{Low}}{N} \right) \left(1 - \left((1 + \alpha_{Hi}) \frac{\rho_{Hi}}{N} + (1 + \alpha_{Low}) \frac{\rho_{Low}}{N} \right) \right)} \quad (13)$$

Using the average latency time (T_{ppp}) between two nodes, we can obtain the packet delay:

$$T_{Hi} = T_{Tx, Hi} + \bar{X}_{Hi} + \alpha_{Hi} T_{TB} + T_{ppp} \quad (14)$$

$$T_{Low} = T_{Tx, Low} + \bar{X}_{Low} + \alpha_{Low} T_{TB} + T_{ppp} \quad (15)$$

3. DELAY CALCULATE

Suppose the packet rounds on a minimum hop, for N nodes RPR, we can obtain the maximum node number that a packet transmission is:

$$\alpha_{Hi/Low}(\max) = \begin{cases} \frac{n-1}{2} & (n \text{ odd}) \\ \frac{n}{2} & (n \text{ even}) \end{cases} \quad (16)$$

The probability of packet arriving each destination is:

$$q_{j, Hi/Low} = \begin{cases} \frac{2}{N-1} & j=1,2,\dots,\frac{N-1}{2} \\ 0, & \text{otherwise} \end{cases} \quad (N \text{ odd}) \quad (17)$$

$$q_{j, Hi/Low} = \begin{cases} \frac{2}{N} & j=1,2,\dots,\frac{N}{2} \\ 0, & \text{otherwise} \end{cases} \quad (N \text{ even}) \quad (18)$$

From formula (3) we can get the average number $_{Hi/Low}$ of TBs:

$$\alpha = \alpha_{Hi/Low} = \frac{N-3}{4} \quad (N \text{ odd}) \quad (19)$$

$$\alpha = \alpha_{Hi/Low} = \frac{N-2}{4} \quad (N \text{ even}) \quad (20)$$

From formula (2), in the store-and-forward architecture, we obtain the service latency time:

$$T_r = \frac{1}{\mu} (1 + \alpha) \left(\frac{\rho_{Hi}}{N} + \frac{\rho_{Low}}{N} \right) \quad (21)$$

Because $\bar{X}_{Hi/Low}$ and T_{pp} are constants, we neglect them. Then we can get the approximate packet transfer delay formulae in store-and-forward architecture:

$$T_{Hi} \approx T_{Tx,Hi} + \alpha_{Hi} T_{TB,Hi} = T_r \frac{1 + \alpha}{1 - (1 + \alpha) \frac{\rho_{Hi}}{N}} \quad (22)$$

$$T_{Low} \approx T_{Tx,Low} + \alpha_{Low} T_{TB,Low} = R \frac{1 + \alpha}{(1 - (1 + \alpha) \frac{\rho_{Hi}}{N})(1 - (1 + \alpha) \frac{\rho_{Hi} + \rho_{Low}}{N})} \quad (23)$$

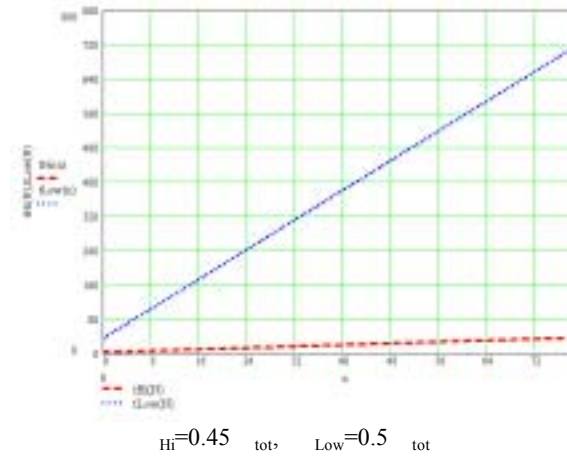
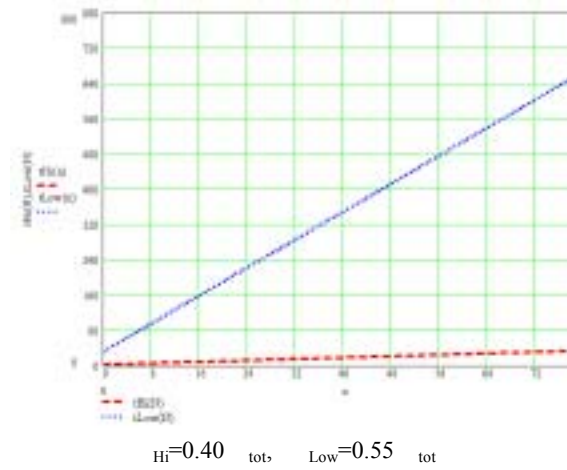
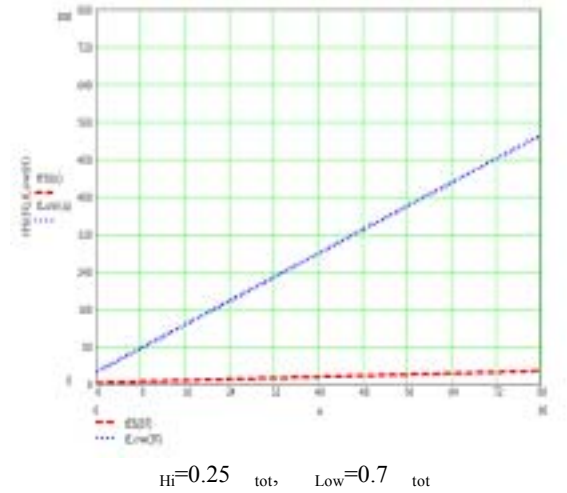
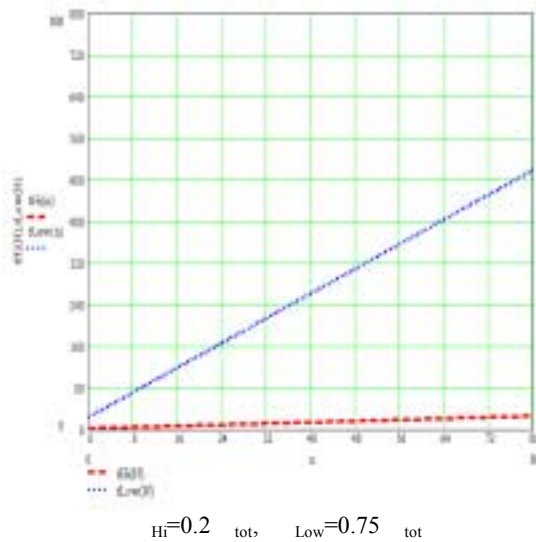
From formulae (30) (31), we obtain the maximum throughput:

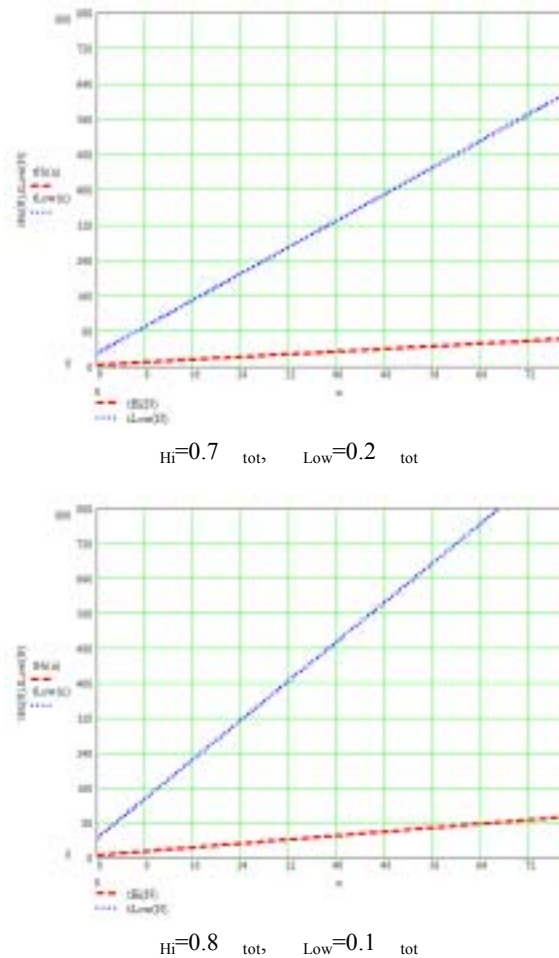
$$\rho_{Hi,Max} < \frac{N}{1 + \alpha} \quad (24)$$

$$\rho_{Low,Max} < \frac{N}{1 + \alpha} - \rho_{Hi} \quad (25)$$

Suppose the maximum throughput $\rho_{tot} = N/(1 + \alpha)$. We discuss the relationship between the packet delay and the node N in a network which has high/low priority traffics.

We simulate some realistic traffic models below. In each traffic models, there are different ρ_{Hi} and ρ_{Low} values. Now we assume: $\rho_{Hi/Low} = \rho_{tot} \times \mu - 1$, $\rho_{Hi} = a \times \rho_{tot}$, $\rho_{Low} = b \times \rho_{tot}$, (a and b have the relationship: $a + b < 1.0$), $\rho_{tot} = N/(1 + \alpha)$ and $N/4$, we can get the 't(N)—N' plot below.





From these “ $t(N)$ - N ” plots, we can see that the packet delay increases (nearly) linearly with the node number N , and the low priority traffic has the larger delay than high priority traffic, and the same as the node number N and different H_i , Low value.

4. CONCLUSION

In this paper, the high and low priority packet transfer delay of the N nodes RPR in store-and-forward architecture is analyzed based on the queuing theory. By deducing we obtain the following results:

- 1) Both high priority and low priority packets' delay increases with the node number N of the RPR rings.
- 2) The high priority traffic has less packet delay than the low priority traffic at the same node number N .
- 3) The increase of the low priority transfer delay is much larger than the high priority traffic with the increase of the node number.

5. REFERENCES

- [1]. “IEEE 802.17 Resilient Packet Ring Working Group Website”, <http://www.ieee802.org/rprsg/>.
- [2]. “An Introduction to Resilient Packet Ring Technology”, a White Paper by the Resilient Packet

Ring Alliance, October 2001.

- [3]. J. L. Hammond and P. J. P. O'Reilly, Performance Analysis of Local Computer Networks, Addison-Wesley, 1988.
- [4]. “Part 17: Resilient packet ring access method and physical layer Specifications”, Submitted to IEEE 802.17 as the Proposal – Darwin , Draft 0.3 , January 14, 2002



Yang Yi is a Ph.D. student of State Key Lab. of Laser Technology, Huazhong University of Science & Technology. She received her M.S. and B.A. degree in computer science from Wuhan University of Technology (WHUT). She is a lecturer of School of Computer Science and Technology of WHUT. Her research interests are in distributed parallel processing, cluster, broadband optical communication, and optical interconnection.

A Kind of Generic Real-time Dependable Server Architecture with Low Fault-latency Using COTS Components

Ou Zhonghong¹, Dai Xingfa¹, Yuan Youguang², Li Haishan¹
¹Institute of Computer Science, Harbin Engineering University
Harbin, Heilongjiang, 150001, China
²Wuhan Digital Engineering Institute
Wuhan, Hubei, 430074, China
E-mail: ouzhongh@sina.com Tel.: +86 (0)27 8753426

ABSTRACT

Most of the highly dependable servers were customized, which resulted in long development period, high cost and made the servers lag behind technology progress and application requirements. To overcome the plight, a significant and feasible solution is to use COTS components including COTS hardware and software, which will substantially shorten development period and lower the cost while obtaining the requested dependability and performance. By designing Intelligent Fault Management Hardware Module and using low latency high-speed proprietary network to parallelize normal operation with fault handling algorithm, the paper puts forward a generic server architecture with low fault-latency GRDS-LFT that is highly dependable, real-time, upgradeable, inexpensive and based on COTS components.

Keywords: Generic, COTS component, fault-tolerant, intelligent, server.

1. INTRODUCTION

Servers play an important role in Internet or network based applications. How to deliver available and reliable services to users is critical. Fault-tolerance is an efficient way to improve server reliability [1], which adopts fault-detection, fault diagnosis, fault isolation and fault recovery mechanism to avoid, reduce and eliminate the contaminations from faults based upon redundancy, such that the server systems can offer available services even in the presence of faults. According to the requested reliability and availability and server system fault models, there were different kinds of fault-tolerant architecture to implement aforementioned fault-tolerant mechanism. In fact, most of the *highly* dependable servers mainly utilized newly-designed hardware or software components, i.e., they were customized, which resulted in long development period, high cost and made the servers lag behind technology progress and application requirements and made them application-specific without any generalization. Though customized fault-tolerance is of high efficiency, its shortages are obvious:

- It may raise fault-tolerant computing cost. The system hardware and the software need to be redesigned, making it unaffordable.
- It is hardly scalable.
- It is difficult to upgrade. Customized design period is so long that its processing capability will not meet users' requirements when it turns out.

To overcome the plight, also to meet different availability and reliability requirements, researchers have been endeavoring to

devise generic fault-tolerant computers [2], [3], [7], i.e., the computers cost is low, scalable, and easy to design and they also keep up with progress of hardware and software. A significant and feasible solution is to use COTS components, including COTS hardware and software, which will substantially shorten development period and lower the cost while obtaining the requested dependability. In fact progress of hardware and software makes the COTS components based realization viable. There are popular processors such as Intel processors that have strong processing power and develop quickly. There are ready COTS computer hardware modules. Communicating and networking technique with low latency and high bandwidth have been developed sufficiently. Free open software such as LINUX operating system is available, and it runs excellently.

The paper is organized as follows: In the following section the works related are summarized by introducing several kinds of fault-tolerant computer architecture utilizing COTS technique in terms of their implementations. Section 3 discusses in detail a kind of generic server architecture with low latency. Finally, Section 4 concludes the paper.

2. WORKS RELATED

2.1 FtServer series [2]

This family of fault-tolerant systems delivers industry-leading uptime of 99.999% and greater for Microsoft® Windows® 2000 applications. Intel® processor-based servers offer high availability while providing operational simplicity and a significant financial advantage. Competitively priced to buy or lease, the ftServer family reduces the initial purchase price of fault-tolerant systems and cuts the costs of ongoing support expenses and unplanned downtime. Because ftServer systems maintain 100 percent compatibility with Windows 2000 Server operating systems at the Application Binary Interface (ABI) level, thousands of off-the-shelf Windows 2000 software products used in today's contact centers have access to the advantages of fault-tolerant server platform. No software modifications or special administrative procedures are needed. Figure 1 shows the architecture of ftServer series. The most important way or foundation to implement high availability of the ftServer architecture is the technology called "Lockstep technology". Lockstep technology uses replicated fault-tolerant hardware components that process the same instructions at the same time. In the event of a component malfunction, the partner component acts as an active spare that continues normal operation and averts system downtime. The system also detects and corrects transient hardware errors that could cause software failures if left unchecked

Summarily, the ftServer features simplicity, software transparency, hardware fault-tolerance and COTS components. But they are also customized. The processors are aboratively selected from Intel and PCI devices fault detection is accomplished by ASICs between CPU and passive backplane and between passive backplane and normal PCI slots. So the ftServer family is the mixture of general technology and specific hardware technology.

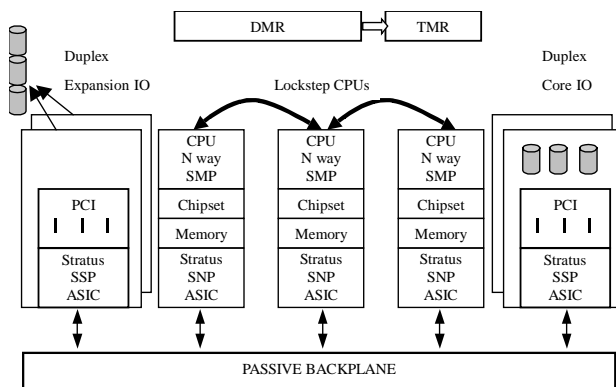


Figure 1. Architecture of FtServer series

2.2 GUARDS [3], [4], [5]

Generic Upgradeable Architecture for Real-Time Dependable Systems (GUARDS) is a project name initiated by Consortium of European companies and academic partners. The project aims to significantly decrease life cycle cost of embedded system, the application domain of which is nuclear subs, railway, and space. Figure 2 is the architecture of GUARDS. The design philosophy is to sufficiently utilize COTS components and multi-layer fault-tolerance and to be open, while using less customized hardware components. Through combination varied dependability requirements are met. GUARDS uses three dimensional fault-tolerance: the channels(C), the first fault containment region, among of which the applications are actively replicated, the lanes (L), the second fault containment region, through which the anomaly of one channel can be detected, and the software integrity (I), the last fault containment region, in which the software design errors can be masked and prevented from spreading.

Obviously, the GUARDS architecture is open, scalable, real-time and upgradeable and with low cost. But the fault latency is too long because the fault detection mechanism mainly depends upon comparison or voting within inter channel network whose latency is considerably long and non-deterministic.

2.3 X2000 project [6], [7], [8], [9]

With NASA's spectacular return to Mars on July 4, 1997, the Mars Pathfinder Lander and its Sojourner Microrover had set a new standard for faster, better, cheaper space exploration missions. Mars Pathfinder was designed as a short-term, low cost mission aimed at demonstrating low-cost entry, descent, and landing on Mars. The aim of a new NASA program, the Advanced Deep-Space Systems Development Program, was to develop (among other things) a highly miniaturized, reliable avionics system for long-life deep-space applications. The near term goal of this program, also known as X2000, was to develop and space-qualify a multi mission engineering model

of the system by 2000. The X2000 avionics system must demonstrate an order-of-magnitude improvement in performance over the Mars Pathfinder system while reducing mass, volume, and power. Because the X2000 avionics system must serve multiple long-life deep-space missions that are also low cost,

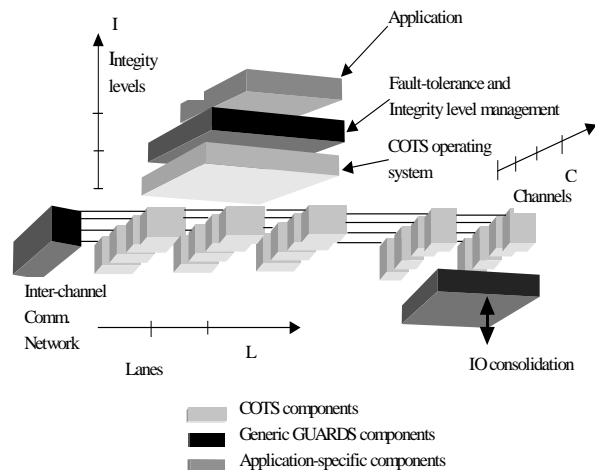


Figure 2. Architecture of GUARDS

its architecture was driven by several key criteria, including reliability and availability, maintainability, evolvability, affordability, and miniaturization. The system uses a distributed fault-tolerant architecture to meet system-level reliability requirements. To dramatically reduce development, integration, and testing costs, X2000 technology required that all spacecraft interfaces, as well as the programming language and software development environment, be based exclusively on commercial off-the-shelf technologies (hardware, software, standards, and so on). NASA used a radiation-hardened COTS-based modular, building-block approach to system design to meet the various multi mission requirements. High-level specification, modeling, and verification were an integral part of system design. Developing highly reliable and long-term survivable missions based on radiation-hardened COTS technologies is a major new strategy for future NASA missions.

3. A KIND OF GENERIC REAL-TIME DEPENDABLE SERVER ARCHITECTURE WITH LOW FAULT-LATENCY USING COTS COMPONENTS (GRDS-LFT)

Low fault-latency is critical in dependable real-time applications [1]. Fault-latency is the time interval between the time a fault occurs and the time the fault is detected and handled. If the latency is too long the fault will contaminate larger part of the system, and the recovery time needed to handle the fault will be considerably long and normal operations will be affected, resulting in that timeliness could not be guaranteed. Fault-detection time and fault-handling time contributes to fault-latency. Software-only-based fault-detection is consequently time-consuming but economical. Hardware-only-based fault-detection is fast enough and timesaving but expensive. The paper presents a kind of architecture, the fault detection and handling mechanism of which is mainly software based but implemented by COTS hardware module.

COTS-components-based highly dependable servers would be unqualified for real-time applications unless the fault-detection mechanism mostly software based only is improved. In traditional COTS dependable servers, e.g. GUARDS, fault-detect algorithms are running in the same CPU as that of operating system and applications. When a checkpoint arrives the only CPU sends the check data and waits to receive and compare. If the normal operations (running OS and applications) are parallelized and pipelined with fault-detection algorithms, fault tolerant overhead can be saved much. In fact another CPU can be introduced to implement the algorithms (this CPU is called FDCPU, i.e., the normal CPUs (this CPU is called host CPU) are parallel with the FDCPU. Moreover, In order to make the FDCPU module generic, the module also confines to COTS components. The primary difference between FDCPU module and the inter channel network (ICN) of GUARDS is that the FDCPU module is intelligent and the fault-detection mechanism is resident within this module.

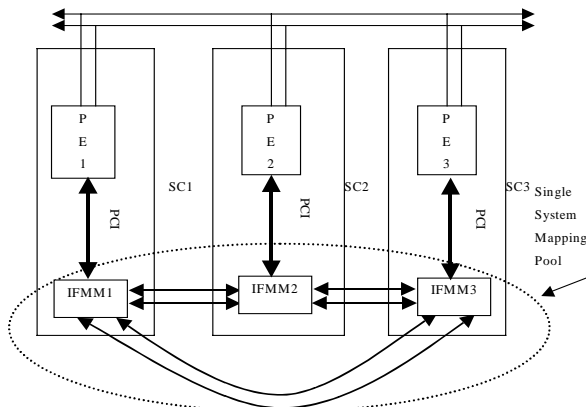


Figure 3. Architecture of GRDS-LFT

As shown in figure 3, N ($N=1$ to 3) homogeneous physical computers (SC1, SC2, or SC3) compose a logical computer (server), each SC includes processing elements PE (host CPU, including one CPU module which supports SMP, dual or single network modules, and etc.) and an intelligent fault management module (IFMM). PE is intra-connected with IFMM through PCI bus, the bandwidth of which is 133MB/s or 533MB/s [10]. All IFMMs are fully connected through dual high-speed (e.g., more than 1000Mbps) proprietary serial links PHN (i.e., there are no single point error).

3.1 Single System Image (SSI)

In the architecture, there is logically a Single System Mapping Pool (SSMP) consisted of 2 or 3 IFMM modules. SSI is maintained by the local IFMM that is fully connected to the other IFMMs. Memory in one IFMM is mapped to that of the

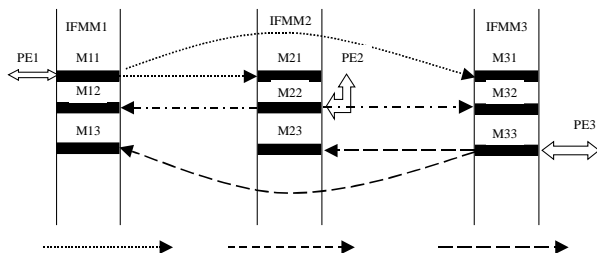


Figure 4. Principle of memory mapping

other IFMMs through the links. The SSMP provides global memory to be accessed by the separate PEs. Figure 4 illustrates the principle of memory mapping.

3.2 Fault-tolerance mechanism

The fault-tolerance mode of GRDS-LFT may be DMR (Dual Module Redundancy) mode or TMR (Triple Module Redundancy) mode. SSMP is the basis of the fault-tolerant server. With the SSMP, synchronization, fault-detection, fault-diagnosis, fault-isolation and recovery can be realized. The 2 or 3 SCs run identical software, as soon as SC encounters checkpoints it sends the checkpoint feature data to local IFMM in the form of specified part of memory shared by IFMM and its host PE, and the local IFMM will simultaneously map the content of the memory to other IFMMs meanwhile receiving check data sent by other IFMMs. One IFMM compares (when number of SC is 2) or vote (when 3) the data copies at the same time when all data copies are received within a pre-assigned time-window. Errors are found and the correct data are used to mask the errors according to the error handling algorithms. When the host PE delivers the checkpoint data it will continue normal operations. The PE can be noticed by its local IFMM when the IFMM has finished checking.

As analyzed above, each SC can be treated as a fault containment region (FCR). Each FCR borders itself by its software and hardware to prevent errors occurring in one FCR from contaminating other FCRs. Faults of varied duration can be tolerated, such as temporary faults, intermittent faults and permanent faults. An FCR will be penalized pre-assigned scores according to the property of errors just occurred. When an FCR reaches the expelling threshold it will be expelled out of normal operation, and vice versa. Figure 5 is the state transfer diagram of FCR or SC.

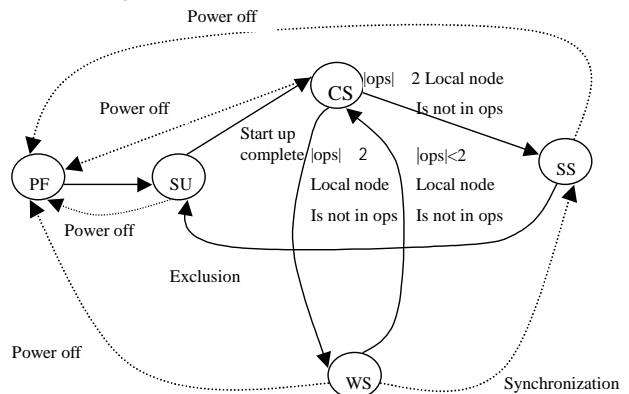


Figure 5. State Transfer of FCR

3.3 Efficiency of fault-tolerance

Fault-tolerance will inevitably bring on time expenses. This is the problem of fault-tolerance efficiency, which is crucial to real-time systems. In order to reduce the time overhead, GRDS-LFT employs several efficient ways. Firstly, PHN is dual links of high speed, without latent collision routes. Consequently, the memory mapping is fast enough. Especially, fault-tolerance algorithm is resident within IFMM module that runs parallel to host CPU. And finally, checkpoints are selected effectively. We can now cursorily evaluate the time consumed for fault-tolerance in GRDS-LFT. Let it be T_{ft} . In fact,

$$T_{ft} = T_1 + T_2 + T_3$$

Where

T_1 is the time used for forming checkpoint data, depending on the data size, and of μ s grade. The size of checkpoint data is less than 100 bytes.

T_2 is the time used for transferring already formed data from host CPU to IFMM via PCI and transferring the data from IFMM to other IFMMs and receiving data from other IFMMs via PHN of 1000Mbps or more. The bandwidth of PCI is 133MB/s or 533MB/s. T_2 is also of μ s grade.

T_3 is the time taken for IFMM to compare and/or vote the received data and to return the result to host CPU, dependent of fault-tolerance algorithm. T_3 is also of μ s grade.

Thus

T_{ft} is of μ s grade.

3.4 Scalability & Upgradeability

Scalability of this architecture consists in its flexibility. Fault-tolerance mode can be DMR or TMR. The availability or reliability is scalable. Due to standard PCI and design-by-module methodology, new function modules can be added in easily. Its structure is scalable.

The hardware and software can be upgraded separately. Moreover, because of independence of fault-tolerance algorithms, the algorithms can be upgraded solely.

4. CONCLUSION

To design real-time dependable servers, COTS components are better choices in terms of affordability, upgradeability, development period and scalability etc. But COTS components based fault-tolerance may be time-consuming if only software based fault-tolerance mechanism is employed. GRDS-LFT is an architecture that is not only affordable, upgradeable, scalable, but also is efficient regarding fault-tolerance. This is achieved by IFMM (COTS component) and low latency PHN that ensures parallelism between normal operations and fault-tolerance operations.

5. REFERENCES

- [1] Yuan Youguang, The Reliability Techniques in Real-Time Systems (in Chinese), Beijing: Tsinghua University Press, September 1995
- [2] STRATUS Technology Corporation. FtServers White Paper, <http://www.stratus.com> USA.
- [3] David Powell, A Generic Fault-Tolerant Architecture for Real-Time Dependable Systems, London: Kluwer Academic Publishers, 2001
- [4] J.Arlat, Preliminary Definition of the GUARDS validation strategy, ESPRIT Project 20716 GUARDS Report, LAAS-CNRS, FRANCE, January 1997
- [5] J.Arlat, et al, Dependability Assessment of GUARDS Instances, IEEE International Computer Performance and Dependability Symposium, USA:IEEE Computer Society Press, 2000, pp.147-156.
- [6] Leon Alkalai, Ann T. Tai, Long-Life Deep-Space Applications, IEEE Computer, vol.31, April 1998, pp. 37-38.
- [7] Ann T. Tai, et al, COTS-Based Fault Tolerance in Deep Space: Qualitative and Quantitative Analyses of A Bus

Network Architecture, Proceedings of the 4th IEEE international Symposium on High Assurance System Engineering, Nov.1999, pp. 97-104.

- [8] Ann T. Tai, et al, On-Board Maintenance for Long-Life Systems, Proceedings of the IEEE Workshop on Application-specific Software Engineering and Technology (ASSET'98), Apr.1998, pp.69-74.
- [9] Savio N. Chau, et al, Design of a Fault-Tolerant COTS-based Bus Architecture, IEEE Transactions on Reliability, vol.48, NO.4, December, 1999, pp.351-359
- [10] Tom Shanley, et al, PCI System Architecture (fourth edition), USA: Addison Wesley Longman, Inc. 1999.



Ou Zhonghong is a doctor candidate of School of Computer Science and Technology, Harbin Engineering University, also a senior engineer of Wuhan Digital Engineering Institute. He graduated in 1989, from Wuhan University with specialty of wireless electronics. His main research interests include fault-tolerance technology, distributed computing and embedded digital system design etc. He has published over 15 papers.



Yuan Youguang, born in 1941, member of IEEE computer society, is a full professor and Ph.D. supervisor of Wuhan Digital Engineering Institute. He received Master Degree from Chongqing University in 1982. He was head of several national advanced research projects and was holder of several national science and technology awards. He has published 2 works and over 150 papers. His main research interests include fault-tolerant computing & reliability theory and distributed computing.

Hardened VPN Based on IXP425*

Yue Hu, Fangmin Li, Quan Liu

School of Information Engineering, Wuhan University of Technology

Wuhan, Hubei province, 430070, China

Email: huyue_whut@hotmail.com Tel: 13907172087

ABSTRACT

The bottleneck of VPN gateway lies on the cryptographic operation consumes massive CPU resources, which directly causes the performance dropping of VPN gateway. According to the conclusion, we propose the implementation of VPN gateway based on Intel's Network Processor, its internal network processing engines provide integrated hardware acceleration for security applications which can support bulk encryption decryption rates up to 70Mbps for DES, 3DES algorithms, thus satisfies the wire speed data processing demand.

Key words: VPN IXP425 encryption

1. INTRODUCTION

In VPN [1] implementation which based on IPSec has used ESP or AH to completes the data protection. But whether uses AH or ESP, they all need to carry on massive data operations, the algorithm includes MD5, SHA-1, DES, 3DES [2, 3] and so on. According to our earlier period research, the VPN gateway speed bottleneck lies in the CPU's disability to perform the encryption. If performing the encryption using software, it will consume massive CPU resource, which inevitably will reduce the VPN gateway's performance.

This article proposed a new method realizing the VPN gateway. This method uses Intel's network processor IXP425. NPEs (Network Processor Engines) in this processor have made hardware optimization to perform the algorithms such as MD5, SHA-1, DES, 3DES and so on, which may reduce CPU's burden, so the VPN gateway can get the wire speed processing ability. This article first points out the bottleneck of VPN gateway based on former experiment result, then gives an introduction to the network processor and its characteristics, after that elaborating how to realize the VPN gateway in detail, finally comes up with the conclusion.

2. DESIGN BASED ON MPC860 AND ITS PROBLEM

In this section, we first introduce our experiment, and then give the conclusion based on the experiment result.

2.1. Hardware Environment

At present, we have completed a prototype system which takes MPC860 as the hardware platform and vxWorks as the operation system. This VPN gateway has a 10BASE-T and a 100BASE-T two Ethernet interface, they are used to connect the protected network and external network.

2.2. System Test Environment

System test take wide band user environment as assumption. Chart 2-1 shows our test environment. In chart 2-1, VPN A and VPN B was two same configured VPN gateway which we already realized. VPN A has connected a subnet with the 100M Ethernet network. The network address is 10.10.1.0/24. Host A is a test machine in this subnet, its IP address is 10.10.1.2/24, default gateway is 10.10.1.254. VPN B has connected in a similar way, its network address is 192.168.1.0/24. Host B is a test machine in this subnet, its IP address is: 192.168.1.2/24, its default gateway is 192.168.1.254. VPN A and VPN B connect with 10M HUB, the IP address respectively are 172.16.1.1/29, 172.16.1.2/29.

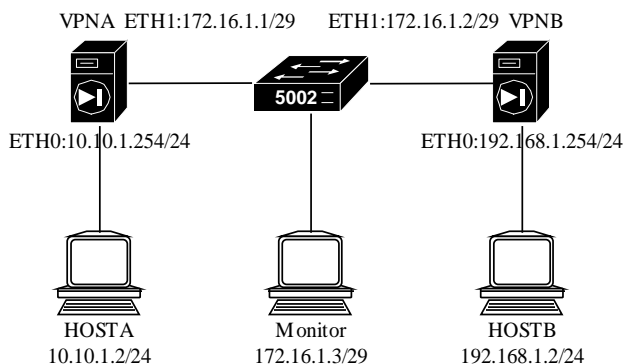


Chart 2-1 Test Topology

2.3. Test result and Analysis

This experiment uses FTP to test the network speed. The concrete test method is that host A runs a FTP server, then use host B to download the same file in different modes. According to transmission time and file size we can get the transmission speed. In order to inspect the speed influence of various protocols, we compare the transmission speed of each protocol with the speed under common condition, thus calculate the ratio. Test data is in table 2-1.

According to data in table 2-1, two protocols do not have much difference when they are processing data in the same way. Take ESP as the inspection object, the ratio using SHA authentication is 0.355, and when using SHA for authentication and DES for encryption, the ratio is 0.150, while using 3DES, the ratio is 0.072. So we can draw the conclusion that whether authentication or encryption has much influence on transmission speed. From the system architecture view, it proved that actually authentication and encryption consume massive CPU resources which cause rapidly dropping of transmission speed. For MOTOROLA's PowerPc processor designed for communication, they are not good at dealing with encryption, so we have to use other method to improve the total performance. We proposed a resolution based on Intel's IXP425 processor and it will be discussed in next chapter.

3. VPN GATEWAY BASED ON IXP425

* Supported by: key project of Ministry of education (key 03120), tackling project of key science and technology of Wuhan city (20021002046)

IXP425 is the next generation network processor designed by Intel, it integrates the processing of data plane, control plane, management plane and application plane on a single-chip. The

Intel IXP425 network processor provides integrated hardware acceleration for security applications. The network processor implements DES, 3DES and AES data encryption algorithms,

Table 2-1 Testing result on MPC860

IPSec Protocol	Encrypt Algorithm	Authentication	High level protocol	File size (KB)	Transmission time (s)	Speed rate	Ratio compare with common condition
None	None	None	FTP	60279	107	676.9	1.000
AH		MD5	FTP	2646	14	189	0.279
AH	None	SHA	FTP	2646	11	240.5	0.355
ESP	None	MD5	FTP	2464	13	203.5	0.300
ESP	None	SHA	FTP	2464	11	240.5	0.355
ESP	DES	MD5	FTP	2464	28	94.5	0.140
ESP	DES	SHA	FTP	2464	26	101.7	0.150
ESP	3DES	MD5	FTP	1270K	56	47.25	0.070
ESP	3DES	SHA	FTP	1270K	54	49	0.072

in addition to SHA-1 and MD5 authentication algorithms which is quite suitable for VPNs. Chart 3-1 shows the internal architecture of IXP425.

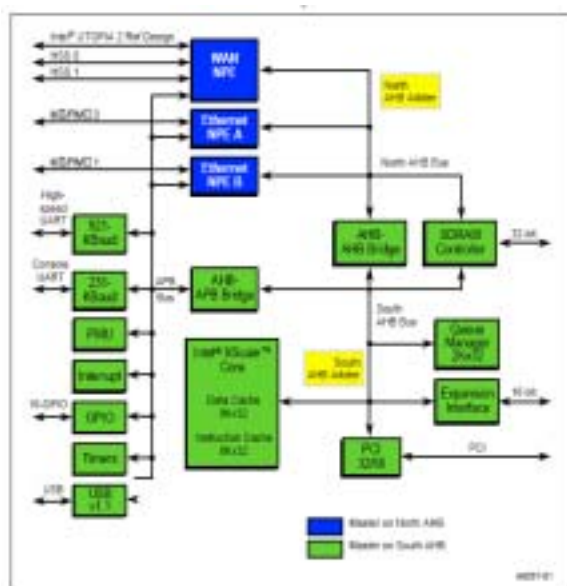


Chart 3-1 Diagram of IXP425

IXP425 uses a distributed structure which makes different operations can be performed in different hardware. We plan to use the XScale core to perform the resource management, while the 3 NPEs are used to deal the transmission data at wire speed. Using the NPE for dedicated hardware acceleration enables the processing of cryptography and authentication algorithms to be offloaded from the Intel XScale core. The high-performance architecture of the Intel IXP425 network processor can support bulk encryption/decryption rates up to 70 Mbps for DES, 3DES and AES algorithms.

Now we will explain our VPN gateway with a typical VPN application, this application is used in the small office or branch office. In local area network aspect, two 10/100M auto-adapted network interface card correspondingly provides

two region's network connection. One region can connect to the exterior network directly which is called the demilitarized zone (DMZ), public resources such WEB server can be set in this area. In this region, data do not need to be protected, data processing in this area will be done by NPE A. The other region is a private area, data transferred in this area should be encrypted, such encryption job will be done by NPE B. Whether the data need to be protected is defined by the corresponding policies, the management of these policies is performed by the XScale core. In WAN aspect, WAN NPE supports many kinds of connections including T1, ATM and frame relaying and so on.

Chart 3-2 is the system logical organization diagram, compares with the original MPC860 implementation, now encrypts and the authentication computation completely done by NPE B which will greatly lighten the CPU's burden. Now CPU's only job is to manage the Security Association Database, Security Policy Database and processing of IKE's negotiation. Using the special-purpose hardware in NPE B to carry on the encryption and authentication operation, its processing speed can achieve 70Mbps.

While using hardware Crypto, we can see from table 3-1 that the performance has improved about three times (data is provided by Intel).

Table 3-1 performance table

Engine Algorithm	Software Crypto	Hardware Crypto
DES, MD5	22~22.3	51.5~52
DES, SHA1	19~19.5	51~51.5
3DES, MD5	9.4~9.5	28.5~29
3DES, SHA1	8.8~8.9	27.5~27.6

In this way, the encryption operation is not the bottleneck of VPN gateway anymore, thus VPN gateway can reach wire speed processing ability.

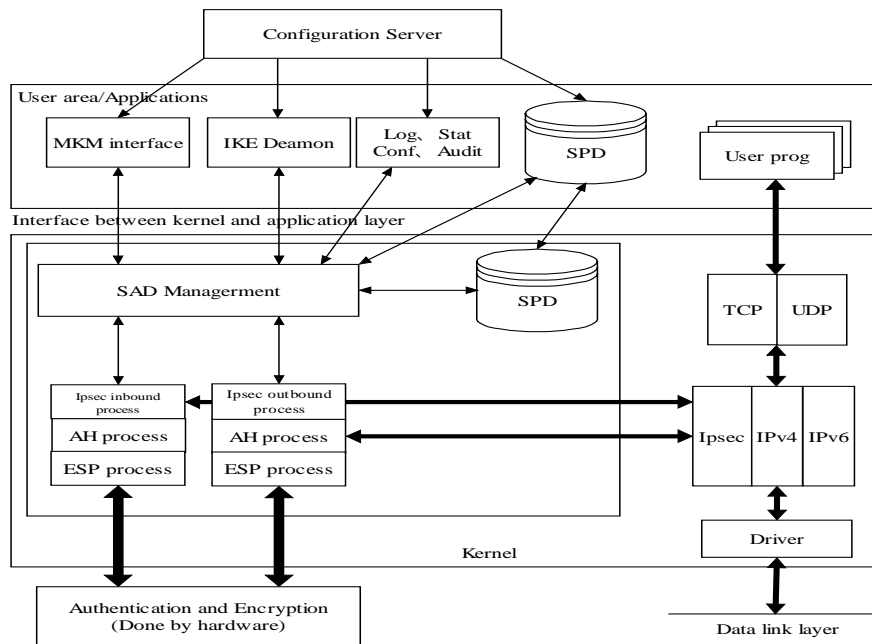


Chart 3-2 Software Framework

4. CONCLUSION

This article analyzes the experiment result of VPN gateway based on MPC860, and points out the key factor influencing the performance of VPN gateway lies on that CPU consumes massive resources to carry on encryption decryption. In order to enhance VPN gateway's performance, we have to use hardware to complete the encryption, releases CPU's resources. We propose to take Intel's network processor IXP425 as prototype, to complete the design of VPN gateway. In this design encryption will be finished by network processing engine and its processing speed can reach 70Mbps, thus may reach the goal of processing data at wire speed.



Fangmin Li, male, born in June, 1968. Doctor, full professor, his research interests are in QoS (Quality of Service), new network protocols, and multimedia communication technology security.

5. REFERENCES

- [1]. IETF. IP Security Protocol (ipsec). <http://www.ietf.org/>, 2003-11-25
- [2]. Diffie W, Hellman M. Multiusers Cryptographic Techniques. Proceedings of the AFIPS National Computer Conference, June 1976
- [3]. Niels Ferguson, Bruce Schneier. A Cryptographic Evaluation of IPsec, Counterpane Internet Security Inc.
- [4]. Intel Corporation. Intel IXP425 Network Processor. <http://www.intel.com/design/network/products/npfamily/ixp425.htm/>, 2003-06-19
- [5]. WRS system. VxWorks 5.5 BSP Developer's Guide. 2002



Yue Hu, male, is a postgraduate in the School of Information Engineering, Wuhan University of Technology. His research interests are in Embedded System, network security.

A New Distributed CFAR Processor

Yang Jun, Ma Xiaoyan, Xiang Jiabin
 Dep. of Information Engineering, Radar Academy
 Wuhan, Hubei, 430010, China
 Email: guoqihu@163.com Tel.: (027) 85965770

ABSTRACT

In the multi-sensor information fusion system, since the centralized CFAR processors bring an overload communication burden, the distributed CFAR becomes a developing and important field. A new OSCA CFAR processor using distributed sensors is presented in this paper. In the scheme, each sensor transmits its test sample and a designated order statistic (OS) of its surrounding observations to the fusion center. At the fusion center, the test samples and the order statistic quantities are combined with the cell average algorithm (CA) respectively to make the final decision. For a Rayleigh fluctuating target in Gaussian noise of unknown level, we obtain its closed-form expressions for the false alarm probability and the detection probability. The numerical results indicate that the detection performance of the proposed OSCA scheme is very close to that of the ideal centralized CFAR, and considerably better than other distributed CFAR processors.

Keywords: Distributed Processor, CFAR, Scaling Factor, Detection Probability, False Alarm Probability.

1. INTRODUCTION

Radar is a typical information system; the detection of radar signal becomes complex when its returns are nonstationary background noise (or noise plus clutter). The probability of false alarm increases intolerably if employing a fixed threshold is used in the detection scheme. Therefore, adaptive threshold techniques are required in order to maintain a nearly constant false alarm rate. Because of the diversity of radar search environment, such as multiple target, abrupt change in clutter, etc, there exists no universal CFAR scheme. A variety of CFAR techniques are developed according to the logic used to estimate the unknown noise power level. Some examples are, the cell-average (CA) CFAR, order statistics (OS) CFAR [1], greatest of CFAR, smallest of CFAR [2], and selection and estimation test [3].

Attraction toward multiple sensor systems with data fusion began to grow in the early 1980s. Distributed signal detection (DSD) schemes are needed when system performance factors such as speed, reliability, and constraint over the communication bandwidth are taken into account. In DSD techniques, each sensor sends either a binary decision or a condensed form of information (statistics) about the observation available at the sensor to the fusion center, where a final decision about the presence of a target is made. DSD with data fusion had been applied to CA CFAR, adaptive CA CFAR[4], and OS CFAR. In these processors, each processor transmits a binary decision to the fusion center where a final decision based on the AND (OSAND) or the OR (OSOR) counting rule is obtained [5, 6]. Instead of a binary decision, each sensor transmits the sample from the test cell and a designated order statistic from available set of reference observations surrounding the test cell to the fusion center, such

a new type distributed processor is considered by Hamid and Viswanathan[7], and they proposed MOS CFAR and mOS CFAR. Numerical results show that the processors outperform those OSAND and OSOR.

On the basis of the MOS CFAR and mOS CFAR We propose a new distributed CFAR detection scheme called OSCA CFAR. Each sensor transmits the sample from the test cell and a designated order statistic (OS) from reference cells to the fusion center. At the fusion center, the test samples and the order statistic quantities are combined with cell average (CA) respectively to complete final decision. In the CFAR processor, the selected order statistics among the sensors could have the same or different ranks, and the number of samples in the reference observations for each sensor need not be the same. At the fusion center, the sum of the test samples is compared with an adaptive threshold obtained by the product of a fixed scaling factor and a function of the received order statistics, to decide whether the presence or the absence of a target.

The performance of the new OSCA processor is compared with the traditional central order statistic (COS) CFAR processor, MOS, mOS OSAND and OSOR processors. Comparison results show that a considerable improvements in performance over the OSAND and the OSOR schemes. Moreover, its performance is close to that of the COS CFAR processor, which has all the test and noise data available.

In section 2, for a N-sensor networks, we research the problem for detecting a Rayleigh fluctuating target in Gaussian noise. Also, closed-form expressions for the probabilities of false alarm and detection for the OSCA processor are derived. Section 3 contains performance comparisons of various schemes based on the numerical study involving a two-sensor network. A summary and the conclusions derived from this study are presented in section 4.

2. DISTRIBUTED OSCA CFAR PROCESSOR

In this section, the OSCA distributed CFAR processor for a network is defined and appropriate parameters are developed. For a two-sensor network, the equations for the processor in homogeneous background are derived.

Consider a N -sensor distributed network as shown in Fig.1. Here, $Y_i = \{Y_{ij}\}$ is observation (excluding the test sample), where $i=1, 2, \dots, N$ indicates the number of the sensors, and $j=1, 2, \dots, N_i$ represents the sample number in the range cells available to the i th sensor. In general, N_{i1} need not be equal to N_{i2} . It is assumed that all the sensors scan the same search environment. The sample in the test cell for the i th sensor denoted by X_{0i} , and the r th rank-ordered adjacent cell observations are denoted by $Y_{i(1)}, Y_{i(2)}, \dots, Y_{i(N_i)}$ where $Y_{i(r)}$ denotes the r th largest order statistics of $\{Y_{i1}, Y_{i2}, \dots, Y_{i(N_i)}\}$. A statistic Z_i from the i th sensor is sent to the fusion center. In our setup, $Z_i = Y_{i(k_i)}$, where k_i is an appropriate integer. At the fusion center, two quantities are computed by the

function $G(\cdot)$ and the function $Z(\cdot)$ respectively, i.e. $G(X_{01}, X_{02}, \Lambda, X_{0N}) = \sum_{i=1}^N X_{0i}$ and $Z(Z_1, Z_2, \Lambda, Z_N) = \sum_{i=1}^N Z_i$. Fusion

center decides the presence or the absence of a target in the test cell by comparing X with TZ , where T is an appropriate scaling factor.

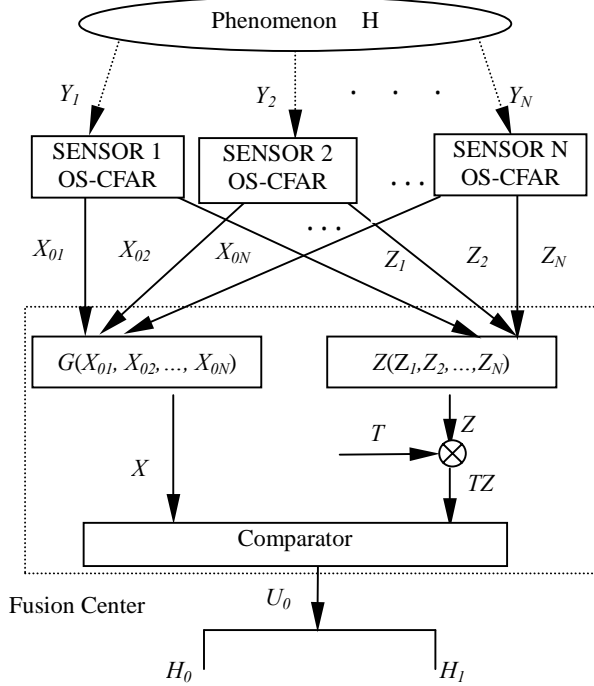


Fig.1 Distributed OSCAR CFAR Processor

It is assumed that $Y_{i1}, Y_{i2}, \dots, Y_{iN}$ are independent identically distributed (I.I.D) random variable that follow an exponential distribution. In the case of homogeneous noise, $E[Y_{ij}] = \lambda_0$ where λ_0 is the noise power and we denote the corresponding density and cumulative distribution function as $f(y)$ and $F(y)$, respectively, let CNR represents the clutter-to noise power ratio. In the case of nonhomogeneous background, the expected value of Y_{ij} is λ_0 or $\lambda_0(1 + \text{CNR})$, depending on whether the sample Y_{ij} is from noise-only region or from clutter, respectively. Assuming a Rayleigh fluctuating target, the test sample, X_{0i} , also has an exponential distribution with mean λ . the mean λ is unknown and depends on the target presence/absence, the clutter level, and the target strength:

$$\lambda = \begin{cases} \lambda_0 & \text{under } H_0 \\ \lambda_0(1 + \text{CNR}) & \text{under } H_1 \end{cases} \quad (1)$$

where hypothesis H_1 represents the presence of a target and hypothesis H_2 means no target, and $\lambda_1 = \lambda_0(1 + \text{CNR})$ represents the signal-plus-noise power, where SNR is the ratio of signal power to noise power. Under H_0 , with clutter background, λ equals $\lambda_0(1 + \text{CNR})$.

For convenience and without loss generality, we study performance of the OSCA processor in the case of two sensors at the fusion center, applying a likelihood ratio test (LRT) to the hypotheses of (1) yields

$$LR = \frac{\prod_{i=1}^2 f_{X_{0i}}(x_{0i} | H_1)}{\prod_{i=1}^2 f_{X_{0i}}(x_{0i} | H_0)} > T_L \quad (2)$$

where T_L is an appropriate threshold. Simplifying (2) yields and achieving the a new CFAR processor is based on

$$X = \sum_{i=1}^2 X_{0i} >_{H_0} TZ(Z_1, Z_2) \quad (3)$$

where T is a scaling parameter that is adjusted to yield a desired false alarm rate under homogeneous background. Since the left-hand side of (3) represents a sufficient statistic of the LRT, the proposed test combines X_{01} and X_{02} in an optimum manner. Because X_{0i} has an exponential distribution, X is a random variable with a gamma distribution whose parameters are 2 and $1/\lambda$, the general form of a gamma probability density function (pdf) with parameter α and β is

$$f(x) = \frac{1}{\Gamma(\alpha)} \beta^\alpha x^{\alpha-1} e^{-\beta x}, \quad x \geq 0, \alpha > 0, \beta > 0 \quad (4)$$

where $\Gamma(\alpha)$ is the gamma function. From (3) we can describe the probability of false alarm P_{fa} as

$$P_{fa} = E_{(Z_1, Z_2)} [P(X \geq TZ(Z_1, Z_2) | H_0)] \quad (5)$$

where $E_{(Z_1, Z_2)}[\cdot]$ represents the expectation with respect to Z_1, Z_2 , hence,

$$\begin{aligned} P_{fa} &= \int_0^\infty (P(X \geq Tz | H_0), Z(Z_1, Z_2) = z) f_{Z|H_0}(z) dz \\ &= \int_0^\infty \left(\int_{Tz}^\infty \frac{1}{\lambda_0^2} x e^{-\frac{x}{\lambda_0}} \right) f_Z(z) dz \end{aligned} \quad (6)$$

where we have used the fact that X and $Z(Z_1, Z_2)$ are statistically independent and that $f_Z(Z) = f_{Z|H_0}(Z)$. We denote the probability of false alarm in the case of homogeneous background noise for OSCA by P_{fa}^{OSCA} . For the OSCA processor, $Z(Z_1, Z_2) = Z_1 + Z_2 = Z$ is the estimate of the noise power of the test cells. We use (6) to derive an expression which indicates the relationship between P_{fa}^{OSCA} and T . The pdf of Z can be expressed as

$$\begin{aligned} P_{fa}^{OSCA} &= \int_0^\infty \left(\int_{Tz}^\infty \frac{1}{\lambda_0^2} x e^{-\frac{x}{\lambda_0}} \right) f_Z(z) dx dz \\ &= \int_0^\infty \left(1 + \frac{Tz}{\lambda_0} \right) f_Z(z) e^{-\frac{Tz}{\lambda_0}} dz \end{aligned} \quad (7)$$

where

$$\begin{aligned} f_Z(z) &= \int_{-\infty}^{+\infty} f_{Z_1}(z-y) f_{Z_2}(y) dy \\ &= \int_{-\infty}^{+\infty} f_{Z_1}(x) f_{Z_2}(z-x) dx = f_{Z_1}(z) * f_{Z_2}(z) \end{aligned} \quad (8)$$

$$f_{Z_1}(z) = k_1 C_{N_1}^{k_1} F_{Y_{1j}}(z)^{k_1-1} (1 - F_{Y_{1j}}(z))^{N_1-k_1} f_{Y_{1j}}(z) \quad (9)$$

$$f_{Z_2}(z) = k_2 C_{N_2}^{k_2} (F_{Y_{2j}}(z))^{k_2-1} (1 - F_{Y_{2j}}(z))^{N_2-k_2} f_{Y_{2j}}(z) \quad (10)$$

and $C_{N_i}^{k_i}$ represents the combination of k_i number from N_i . Substituting (9) and (10) into (8) and using equations $f_{Y_{ij}}(y) = e^{-y}$, $f_{Y_{ij}}(y) = 1 - e^{-y}$ yields

$$\begin{aligned} f_Z(z) &= \int_{-\infty}^{+\infty} k_1 k_2 C_{N_1}^{k_1} C_{N_2}^{k_2} (F_{Y_{1j}}(x))^{k_1-1} (F_{Y_{2j}}(z-x))^{k_2-1} \\ &\quad (1-F_{Y_{1j}}(x))^{N_1-k_1} f_{Y_{1j}}(x) (1-F_{Y_{2j}}(z-x))^{N_2-k_2} f_{Y_{2j}}(z-x) dx \\ &= k_1 k_2 C_{N_1}^{k_1} C_{N_2}^{k_2} \int_{-\infty}^{+\infty} (F_{Y_{1j}}(x))^{k_1-1} (F_{Y_{2j}}(z-x))^{k_2-1} \\ &\quad (1-F_{Y_{1j}}(x))^{N_1-k_1} (1-F_{Y_{2j}}(z-x))^{N_2-k_2} f_{Y_{1j}}(x) f_{Y_{2j}}(z-x) dx \\ &= A \int_0^{+\infty} (1-e^{-x})^{k_1-1} (e^{-x})^{N_1-k_1+1} \\ &\quad (1-e^{-(z-x)})^{k_2-1} e^{-(z-x)(N_2-k_2+1)} dx \end{aligned} \quad (11)$$

where $A = k_1 k_2 C_{N_1}^{k_1} C_{N_2}^{k_2}$, without loss its generality, we can set $\lambda_0 = 1$, and can rewrite (7) as

$$\begin{aligned} P_{fa}^{OSCA} &= \int_0^{\infty} (1+Tz) f_Z(z) e^{-Tz} dz \\ &= \int_0^{\infty} e^{-Tz} f_Z(z) dz + T \int_0^{\infty} z e^{-Tz} f_Z(z) dz \end{aligned} \quad (12)$$

Using the identities

$$\int_0^{\infty} (e^{-x})^{\alpha} (1-e^{-x})^{\beta} dx = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta+1)} \quad (13)$$

$$\int_0^{\infty} x(e^{-x})^M (1-e^{-x})^L dx = L \sum_{i=0}^N \frac{P_L^i}{P_{M+i}} \frac{\Gamma(M+i)\Gamma(L-i+1)}{\Gamma(M+L+1)} \quad (14)$$

where α, β are positive integers, and P_N^k denotes the permutation number of k from N . After performing the appropriate integration and straightforward simplifications, we obtain

$$P_{fa}^{OSCA} = ACE + ATDE + ATCF \quad (15)$$

where

$$C = \frac{\Gamma(k_2)\Gamma(N_2-k_2+1+T)}{\Gamma(N_2+1+T)} \quad (16)$$

$$E = \frac{\Gamma(k_1)\Gamma(N_1-k_1+1+T)}{\Gamma(N_1+1+T)} \quad (17)$$

$$D = \sum_{i=0}^{k_2-1} \frac{P_{k_2-1}^i}{P_{M_2-k_2+1+T+i}^{i+1}} \frac{\Gamma(k_2-i)\Gamma(N_2-k_2+1+T+i)}{\Gamma(N_2+T+1)} \quad (18)$$

$$F = \sum_{i=0}^{k_1-1} \frac{P_{k_1-1}^i}{P_{N_1-k_1+1+T+i}^{i+1}} \frac{\Gamma(k_1-i)\Gamma(N_1-k_1+1+T+i)}{\Gamma(N_1+T+1)} \quad (19)$$

Therefore, we can use equivalent form of (15) to evaluate the probability of false alarm of the processor.

For calculating the probability of detection of the processor, we replace T with $T/(1+SNR)$ in (15), i.e.

$$P_d^{OSCAR} = P_{fa}^{OSCAR} \Big|_{T=\frac{T}{(1+SNR)}} \quad (20)$$

3. NUMERICAL RESULTS

In this section we discuss the numerical results obtained from an evaluation of the performance equation of the OSCA processor, and compare it with classical processors such as the central order statistic processor, the distributed CFAR and the AND, and the OR processor. For a two-sensor network, our numerical analysis is carried out for the specific values of various parameters, which be listed in Table I.

Table 1 Parameters and Calculated Values for Constant T for $P_{fa}=10^{-6}$

Processor		Refer-ence Cells	Rank	P_{fa}	T
COS	Sensor 1	11	17	10^{-6}	23.64
	Sensor 2	13			
OSOR	Sensor 1	11	8	5×10^{-5}	$T_1=36.26$
	Sensor 2	13	9	5×10^{-5}	$T_2=34.23$
OSAND	Sensor 1	11	8	10^{-3}	$T_1=9.555$
	Sensor 2	13	9	10^{-3}	$T_2=9.725$
MOS	Sensor 1	11	8	10^{-6}	21.96
	Sensor 2	13	9		
mOS	Sensor 1	11	8	10^{-6}	47.30
	Sensor 2	13	9		
OSCA	Sensor 1	11	8	10^{-6}	12.34
	Sensor 2	13	9		

The detection performance of all CFAR processors, in the homogeneous background noise, are shown in Fig.2, it is obviously seen that the OSCA nearly overlap with COS processor. Fig.3 shows that the difference between OSCA processor and COS processor, that is to say, OSCA can have a better detection performance comparing with OSAND, OSOR, MOS and mOS processors.

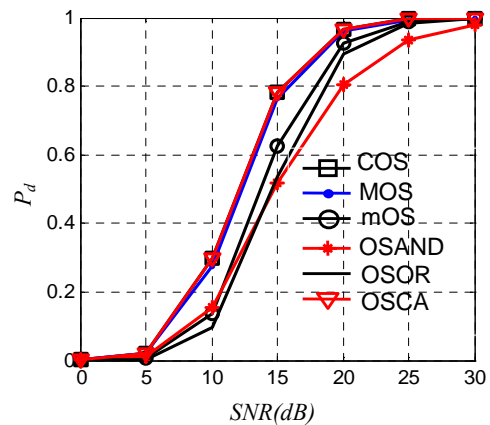


Fig.2 probability of detection versus SNR when background noise is homogeneous

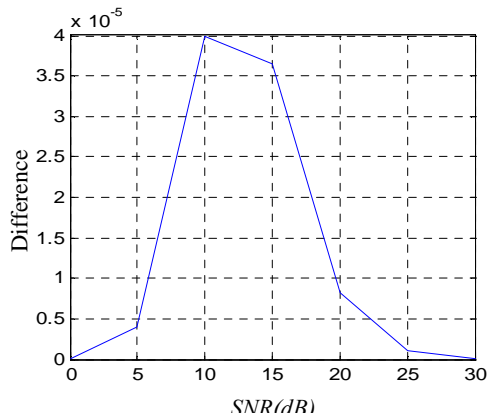


Fig.3 difference of probability of detection between COS and OSCA processors

4. SUMMARY AND CONCLUSION

In this paper, we have developed a new OSCA CFAR processor used to distributed sensors system. Our problem formulation has assumed that the test cells of different sensors all have statistically identical noise (clutter), and that if a target is present in the surveillance regions, all the test cells have statistically identical target returns. This requirement implies that all sensors have the same test SNR. In the OSCA scheme, each sensor transmits its test sample and a designated order statistic of its surrounding observations to a fusion center, where the sum of the samples of the test cells is compared with the sum of the order statistics of each sensor. For detecting a Rayleigh fluctuating target in Gaussian noise, closed-form expressions for the false alarm and the detection probability are obtained. And general equations for the false alarm probability under homogeneous background noise presented. Performance of the scheme is compared with the OSAND CFAR processor, the OSOR CFAR processor, the MOS CFAR processor, the mOS CFAR processor and the COS CFAR processor.

We conclude from the study of two-sensor network that for the homogeneous background noise, the detection performance of the proposed OSCA scheme is very close to that of the COS CFAR, and considerably better than those of the MOS CFAR, mOS CFAR, OSAND CFAR and OSOR CFAR.

5. REFERENCES

- [1]. H. Rohling, "Radar CFAR Thresholding in Clutter and Multiple Target Situation", IEEE Transactions on Aerospace and Electronic Systems, Vol.19, No.2, JULY 1983, pp.608-621.
- [2]. P. Gandhi, S.A. Kassam, "Analysis of CFAR Processors in Nonhomogeneous Background". IEEE Transactions on Aerospace and Electronic Systems, Vol.24, No.2, JULY 1988, pp.427-445.
- [3]. R. Viswanathan, A. Eftekhari, "A Selection and Estimation Test for Multiple Targets in Clutter Detection". IEEE Transactions on Aerospace and Electronic Systems, Vol.28, No. 1, APRIL 1992, pp. 505-519.
- [4]. M. Barkat, P.K. Varshney, "Adaptive Cell-averaging CFAR Detection in Distributed Sensor Networks". IEEE Transactions on Aerospace and Electronic Systems, Vol.27, No.2, MAY 1991, pp. 424-429.
- [5]. M. K. Uner, P. K Varshney, "Decentralized CFAR Detection Based on Order Statistics", In Proceeding of 36th Midwest Symposium on Circuits and systems, Detroit, MI, August 1993, pp. 146-149.
- [6]. R. S. Blum, S. A. Kassam, "Distributed Cell-averaging CFAR Detection of Dependent Signal Returns", Proceeding of 1993 IEEE International Symposium on Information Theory, January 1993, pp.12-15.
- [7]. A. Hamid, R. Viswanathan, "A New Distributed Constant False Alarm Rate Detector", IEEE Transactions on Aerospace and Electronic Systems, Vol.33, No.1, January 1997, pp. 85-97.

A Scalable Approach for IP-Multicast in Differentiated Services Networks

Wang Xiaoyan, Zheng Mingchun

Department of Computer science, Shandong Normal University

Jinan , Shandong, China, 250014

Email: wangxiaoyan@sdre.net.cn Tel.: 13064045436

ABSTRACT

The phenomenal growths of group communications and QoS-aware applications over the Internet have respectively accelerated the development of two key technologies, namely, multicasting and Differentiated Services (DiffServ). The integration of multicasting support in the DiffServ domain is useful in several aspects, however, the conflicts between traditional multicast and DiffServ make the integration of the two technologies a nontrivial task. This paper demonstrates some of the problems which will arise when IP Multicast is used in DiffServ networks without taking special precautions into account for providing it. Those problems mainly lead to situations in which other service users are affected adversely. In this paper, we propose a simple and scalable approach to retain the benefits of the DiffServ architecture in multicast, and give some simulation results. Finally, we present the future work to ameliorate this approach.

Keywords: QoS-aware, IP-multicast, DiffServ, simulate

1. INTRODUCTION

Recently, Internet services offering a better quality than the current deployed best-effort service are urgently required. Many advanced applications need certain QoS assurances from the network layer, e.g., a maximum delay, a minimum packet loss rate or guaranteed transmission rate. In order to provide QoS to users across the Internet, there are two different thoughts for providing QoS. The first is to increase the bandwidth available to users such that the extra capacity of the network allows all users to meet their appropriate QoS. In contrast, the second is that bandwidth can never be unlimited and therefore the limited bandwidth should be appropriately prioritized among users. The IETF attempted to meet these trends in defining the Integrated Services (IntServ)[1] architecture, which provided quality based services even for group communication scenarios in the Internet. However, the IntServ Architecture shows some inherent scalability problems if applied Internet-wide, especially within backbone areas. Because service differentiation in the Internet was and is still required, the Differentiated Services (DiffServ) [2] Architecture was developed to overcome these scaling problems. Scalability is achieved by avoiding complexity and maintaining per-flow state information in core routers and pushing unavoidable complexity to the network edges. Therefore, individual flows belonging to the same service are aggregated, thereby eliminating the need for complex classification or managing state information per flow in interior routers.

Apart from QoS assurances, another important aspect of the Internet usage is bandwidth utilization. Several evolving applications like WWW, video/audio on-demand services, and teleconferencing consume a large amount of network bandwidth. Multicasting is a useful operation for supporting

such applications. Using the multicast services, data can be sent from a source to several destinations by sharing the link bandwidth (i.e. the number of receivers increases without the traffic on the network increasing remarkably). By reducing the information being transmitted across the network, multicast essentially increases the QoS given to other users of the network due to the additional bandwidth in the network.

From an initial glance, IP multicast and DiffServ are complementary technologies. But the reduced complexity in routers makes it more complex to provide DiffServ together with IP Multicast without any addressed details of DiffServ multicast services yet[2,5].

1.1. Differentiated Services

The DiffServ model contains two types of routers, edge routers and core routers (cf. Fig. 1). Core routers are relatively simple routers designed for the purpose of high-speed routing over the network backbone. Core routers do not maintain any per-flow state information and schedule the packets as per the DSCP [3] within each packet. Thus, the "intelligence" in the DiffServ network is migrated to the edge of the network at the edge routers. The edge router is the key element for proper functioning of the DiffServ network. Responsibilities of the edge routers include proper marking of non-DiffServ-aware traffic, traffic policing, and traffic shaping. It is the responsibility of the edge routers to maintain proper traffic levels to achieve QoS differentiation in the network core.

In the DiffServ Architecture services are constructed from per-hop behaviors (PHB) and some related traffic conditioning actions (e.g., metering, marking, shaping or dropping) which are applied to packets along their path. A packet is usually classified and marked to receive a particular forwarding behavior in the first DiffServ-capable node (i.e. the 'First-Hop Router', cf. Fig. 1) along its path according to their corresponding traffic profile that is selected by the classifier. The forwarding behavior that a packet experiences is identified by the codepoint in the IP packet header. Each codepoint (DSCP) is a specific value conveyed by the Differentiated Services Field (DS-field) that replaces the part of the common Type of Service (ToS) field in IPv4 packets and the class field in IPv6 packets. Different PHBs may use distinct queuing mechanisms in order to achieve the intended differential forwarding treatment of packets.

Packets on the same link in a particular direction carrying the same codepoint are denoted as Behavior Aggregate (BA). After initial setting of the codepoint, subsequent nodes on the path typically operate only on those aggregates. Therefore, 'core routers' (cf. Fig. 1) only have to classify packets by their specific codepoint and treat them with the corresponding forwarding mechanism. Therefore, this model is highly scalable, because they do not have to keep and maintain per flow-states and reservation information.

It is important to notice that changing a codepoint for packets of a particular flow in the interior network requires usually per-flow classification, thus leading to the same scalability problems which the IntServ approach possesses.

1.2 IP-Multicast Fundamentals

The advanced group communicating applications demanding certain QoS assurances from the network layer are stirring the need for efficient multicast communication services satisfying the QoS requirements of those applications [6,7]. There are two ways by which group communication can be achieved: multiple unicasts and multicast. IP multicast can reduce the demand of bandwidth than unicast significantly when transmitting same content to the same user group (cf. Fig. 2). Multicast and DiffServ are different in three aspect. First, the multicast tree requires per-group information at each core router in terms of the routing table entries whereas DiffServ is status-less in core routers. Second, the multicast tree goes against the principle of the DiffServ model in which each core router is independent of other core routers. Last, one-packet-in may translate to many-packets-out because of the packet replications in the multicast tree and thus monitoring the quantity of traffic from a source becomes an arduous task.

While the basic DiffServ mechanisms work also for multicast packets because of packets carrying the DSCP which determines the forwarding treatment (cf. Fig. 1), supplying DiffServ multicast services is not straightforward [2,3]. And we assume, unless stated otherwise, our discussion based on the point-to-multipoint communication scenario but multipoint-to-multipoint one.

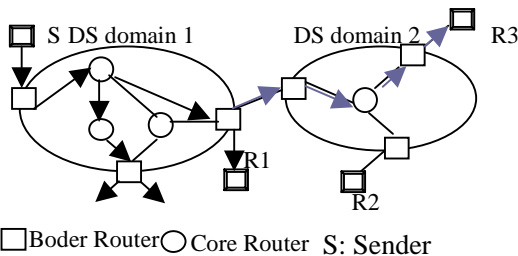


Fig1: Example of two DiffServ Domains using IP Multicast

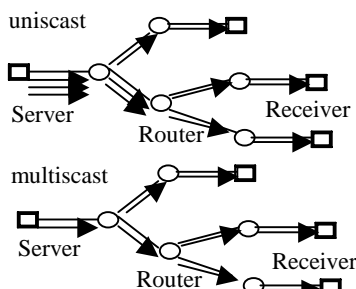


Fig. 2 Unicast and multicast

2 PROBLEMS TO PROVIDE MULTICAST IN DS DOMAIN[2,7]

The simplicity of the DiffServ Architecture and its router models is necessary to reach high scalability, but it causes also fundamental problems in conjunction with the provision of IP Multicast in DS domains. Resources for some DiffServ

services must be reserved before they will be actually used. But providing suitable resources is difficult, because receivers can dynamically join and leave an mc-group anytime, thereby leading to a dynamic resource consumption. If this fact is not considered, it will lead to the problem described in the following section.

2.1 Neglected Reservation Subtree Problem

IP Multicast packet replication takes place when the packet is handled by the routing process. Thus, a DiffServ capable node would also copy the contents of the DS field [2] into the IP packet header of every replicate, and, therefore experience the same forwarding treatment as the incoming packets of this mc-group. Thus the currently provided QoS level of other receivers (with correct reservations) will be adversely affected or violated. This negative effect is the so-called Neglected Reservation Subtree Problem (NRS Problem) .

One can distinguish two distinct major cases of the NRS Problem:

Case 1 — If the branching point of the new subtree and the previous mc-tree is an egress edge router, the additional mc-flow increases the amount of used resources for the corresponding aggregate and will be greater than the originally reserved amount. Consequently, the policing component in the egress edge router discards packets until the traffic aggregate is conforming to the traffic contract. But the discarding is random, whether they belong to a correctly reserved flow or not. As a result, there will be no longer any service guarantee for the reserved flows.

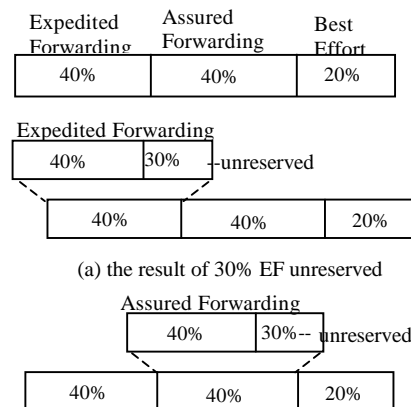


Fig. 3 the initial shared bandwidth and the case1 of the NRS problem

As presented in the Fig3, one link is shared by 40% EF aggregates, 40% AF ones and 20% BE. If additional unreserved 30% EF[9] or AF[10] aggregates are sharing the same link, the share of every class of aggregate is presented in (a) and (b). We can see that unreserved traffic stole the initial share of the same class.

Case 2 — The NRS Problem can also occur when the new subtree is located in an interior router. Because the router is usually not equipped with metering or policing functions it will not recognize any excess amount of traffic and will forward the new mc-flow. If the latter belongs to a higher priority service, such as EF, bandwidth of the aggregate is higher than the aggregate's reservation and it will steal bandwidth from lower priority services. The additional

amount of EF without a corresponding reservation is forwarded together with the aggregate that has a reservation. This results in no packets losses for higher priority as long as ones with lower priority are not discarded completely.

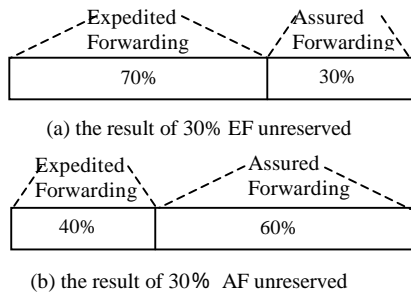


Fig. 4 Case2 of the NRS problems

In the case2 with the same conditions, we can see from the Fig4 that unreserved traffic stole the share(s) of the class (es) whose priority (ies) is(are) lower.

2.2 Dynamics of Arbitrary Sender Change

Because Differentiated Services are unidirectional by definition, we also consider the point-to-multipoint communication being of unidirectional nature. However, in traditional IP Multicast any node can send packets spontaneously and asynchronously to an mc-group, respectively to its multicast group address. Therefore, for every mc-tree implied by a sender resources must be reserved separately if simultaneous sending should be possible with a better service. This is even true if shared multicast delivery trees are used (e.g., with PIM-SM or Core Based Trees). Unless single-source are used, there is no possibility to ensure in the IP layer that only one sender is transmitting at the same time. Otherwise, the NRS problem will occur again.

2.3 Heterogeneous Multicast Groups

Multicast groups may contain one or more receivers, which would like to get another service or quality of service as the sender provides or other receiver subsets currently use. A very important characteristic which should be supported by Differentiated Services is that participants requesting a best-effort quality only should also be able to participate in a group communication which otherwise utilizes a better service class. The next better support for heterogeneity provides concurrent use of more than two different service classes within a group. Because an arbitrary new receiver that wants to get the different service can be grafted to any point of the current multicast delivery tree, even interior nodes may have to replicate packets using the different service. This seems to be a contradiction with respect to simplicity of the interior nodes, because they do not even have any profile available and should now convert the service quality of individual receivers.

3 APPROACH PROVIDE IP-MULTICAST IN DIFFERENTIATED SERVICES NETWORKS

The problems described in the previous sections are mainly caused by the simplicity of the DiffServ Architecture. Solutions have to be developed without introducing an additional complexity that diminishes the scalability of this approach. An architecture is suggested to provide simple solutions for the described problems

3.1 The Main Idea of Our Approach [7,8,11]

The proposed solution consists conceptually of the following two steps that are described in more detail later.

1. A new receiver joins a multicast group that is using any DiffServ service. Multicast routing protocols accomplish the connection of the new branch to the (possibly already existing) multicast delivery tree as usual if the join is available to provide the requested QoS, the only difference is that the join must be an edge router.
2. Choose the first-respond edge router if there is no edge router satisfy the join request, and the unauthorized use of resources is avoided by re-marking at branching nodes all additional packets leaving down the new branch, i.e. the new receiver will get all packets of the multicast group without QOS guarantee. When the pre-issued reservation is available for the new branch, the management entity instructs the branching router to set the corresponding codepoint for the demanded service.

3.2 The Detailed Implement of the Approach

In the following discussed example, the case is considered when the join of a new receiver to a DS multicast group requires grafting of a new branch to an already existing multicast delivering tree. The join process is triggered by the receipt of a multicast join message from a new receiver (using IGMP or other signal protocols). When an edge node (E_{Rcvr} , the closest edge node to the new receiver which connect the new receiver to the tree) wishes to join to a multicast group, the E_{Rcvr} sends a Join-Request to all eligible edge routers via a pre-constructed control tree (i.e. E_{Rcvr} must graft to an edge router, it's reasonable because the core router is stateless and it can provide favorable scalability), then the E_{Rcvr} sets a timer to be triggered Bid-Wait seconds to evaluate the bid responses of the valid edge nodes (the Bid-Wait timeout value is selected by the network administrator and configured at each edge router).

The edge router which receives the Join-Request will send a Join-Bid message. The bid message consists of routing information (populated by edge router) and dynamic information (populated by core router). And the routing information can be divided into two classes, one is sent from the source or the ingress router of the domain which already has a tree constructed for the group, thus the E_{Rcvr} just need to add the new receiver to the multicast tree. Alternatively, the edge router without an existing distribution tree for the multicast group will set up a tree with E_{Rcvr} as its only node. In addition, the edge node may have to do inter-domain routing.

As E_{Rcvr} receives the Join-Bid message, it is responsible for processing the messages to determine the best edge router to join for the multicast group depending on the requested QoS. We also divide the responses of the Join-Bid messages into two classes: one for the situation having available router to graft the E_{Rcvr} (i.e. the new receiver) which is sent before

timeout (the timer is mentioned above), and the E_{Rcvr} will chose to graft to the edge router which response first i.e. the first-come the first-choose. Thus the new receiver will incept the replications just as the previous receivers.

And, the another type of Join-Bid message for the situation having no available router to satisfy the request of the new receiver which is sent after timeout. The E_{Rcvr} will chose to graft to the edge router sent the first-come message and the E_{Rcvr} converts the codepoint to a codepoint of a PHB which is similar to the default PHB in order to provide a best-effort-like service for the new branch. More specifically, this particular PHB can provide a service that is even worse than the best-effort service of the default PHB. Furthermore, the re-marked packets from this multicast group should be discarded more aggressively than BE aggregate. This could be accomplished by using the Limited Effort (LE) PHB [4].

The conversion to this specific PHB could be necessary in order to avoid unfairness being introduced otherwise within the best-effort service aggregate, and, which results from the higher amount of resource usage of the incoming traffic belonging to the multicast group when the resources are scarce.

Re-marking packets is only required at E_{Rcvr} , whereas all other nodes of the multicast tree replicate packets as usual. And, the better service will only be provided if a reservation request was processed and approved by the resource management entity (Bandwidth Broker-BB for example). In case the admission test is successful, the re-marking node will be instructed by BB to stop re-marking and to set corresponding codepoint for the demanded service.

Therefore, in a DiffServ multicasting group, only those receivers will obtain a better service, which previously reserved the corresponding resources in the new branch. Otherwise they will get the quality which might be even lower than best-effort.

3.3 Simulation Results

We do some simulations under NS2 with an simple topology (cf. Fig. 5).

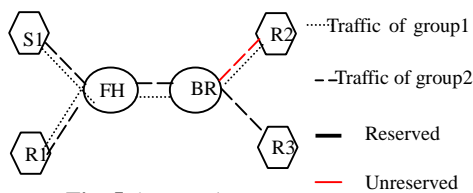


Fig. 5 the Topology

The S1 generates two shaped UDP traffic flows of 500 kbps (packets of 1000 byte constant size) each and sends them to multicast group 1 (234.1.1.1) and 2 (234.1.1.2). In both measurements R1 has a reservation along the path to the sender for each flow, R2 has reserved for flow 1 and R3 for flow 2. And, each link's bandwidth is 10Mbps, thus, there is no bottleneck in this topology. Therefore, two static profiles are installed in the first-hop router with 500 kbps EF and a token bucket size of 10000 byte for each flow. In the egress edge router one profile has been installed for the output link to R2 and one related for the output link to R3. Each of them permits up to 500 kbps EF, but only the EF aggregate carried on the outgoing link is considered.

Fig6 presents that the case1 of NRS problem without considering the dynamic join of the new receiver even though the bandwidth is sufficient, and Fig7 presents the throughputs using the proposed approach, and we can see that there is no NRS problem under the same situation.

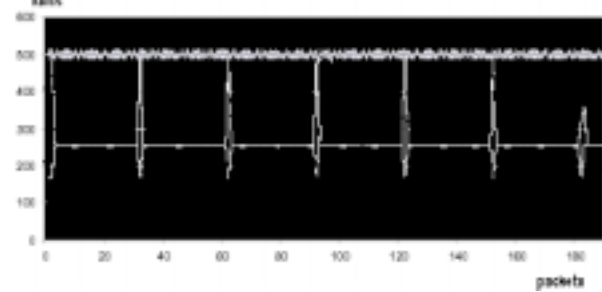


Fig. 6 Throughputs without any precautions
(i.e. DiffServ multicast with NRS problem)

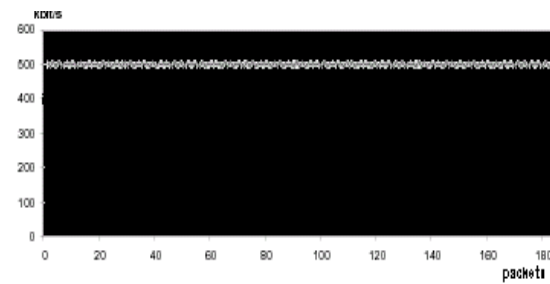


Fig. 7 Throughputs under proposed approach

4. CONCLUSIONS

In this paper, we proposed an protocol-independent approach for providing scalable DiffServ multicast. In this paper, two fundamental multicast provisioning problems were identified: resource usage conflicts due to neglected reservations as well as the scalability in the DiffServ Multicast. The proposed scalable and efficient solution uses an two-step way, to graft to the first responded edge router and incept the replications normally if there is available edge node to provide the requested QoS or to graft to the first-respond edge node and receive the replications with priority even lower than the best-effort service of the default PHB. In the future, a more efficient solution for choosing the appropriate edge node in the second step has to be developed. Furthermore, we have confined our approach in the point-to-multipoint communication scenario, how to expand it to the multipoint-to-multipoint circumstance is a significative work.

5. REFERENCES

- [1]. R.Braden, D. Clark, and S. Shenkar, "Integrated Services in the Internet Architecture: An Overview", RFC 1633, IETF, June 1994.
- [2]. S. Blake et. al, "An Architecture for Differentiated Services", RFC 2475, IETF, Dec. 1998.
- [3]. F. Baker, D. Black, S. Blake, and K. Nichols. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, Dec. 1998.
- [4]. R. Bless and K. Wehrle. A Limited Effort Per-Hop Behavior, Internet-Draft-raft-bless-diffserv-le-phb-00.txt,

- Feb.2001. Work in progress.
- [5]. A. Striegel, G. Manimaran, "A Scalable Approach for DiffServ Multicasting", Proc. of International Conference on Communications, Helsinki, Finland, June 2001.
 - [6]. K. C. Almeroth, "The Evolution of Multicast," IEEE Network, pp. 10-21, Jan./Feb. 2000.
 - [7]. R. Bless, K. Wehrle, "IP Multicast in Differentiated Services Networks", IETF Internet Draft draftbless-diffserv-multicast-07.txt, Aug. 2003.
 - [8]. C-Y Lee, N. Seddigh, "Controlling the number of egress points in dynamic multicast groups", IETF Internet Draft draft-leecy-multicast-egress-limit-00.txt, Oct. 1999.
 - [9]. V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forwarding PHB. RFC2598, June 1999.
 - [10]. F. Baker, J. Heinanen, W. Weiss, and J. Wroclawski. Assured Forwarding PHB Group. RFC 2597, June 1999.
 - [11]. "Multicast Source Discovery Protocol", Cisco IP Multicast Training Materials, 2000.



Wang Xiaoyan is a graduated student of Computer Science and Technology, Shandong Normal University. She graduated from Shandong normal University in 2002; Her research interests are in congestion control in multicast, grid computing, network QoS and TCP/IP protocols betterment.

Zheng Mingchun is a Full Professor and a dean of Computer Science and Technology, Shandong Normal University of Technology. She graduated from Shandong Normal University in 1986; from East China Normal University of Science and Technology in 1992 with specialty of parallel processing. Her research interests are in distributed parallel processing, Heterogeneous Network, network QoS and unicast/multicast congestion control.

A Virtual Server Scheme Based on LVS and XRN *

Guo Yucheng, Guo Qingping

School of Computer Science and Technology

Wuhan University of Technology, Yu Jia Tou Campus, Wuhan, China, Post Code 430063

Email: virusars@163.com

ABSTRACT

Since traffic on the Internet is increasing rapidly, servers with high scalability and reliability are critically needed for enterprises. The paper proposes a virtual server scheme based on a new hardware technology, the XRN, and the software package, the Linux Virtual Server, to construct a high performance and high scalability Server Cluster to release pressure the explosive growth of the Internet brings us.

Key words: LVS, XRN, Virtual Server, Server Cluster, Load Balancer.

1. INTRODUCTION

The Internet became the hottest and the most important issue in Information Technology during these years. At the same time, the traffic on the Internet is increasing dramatically, which has been growing at over 100% annual rate. The workload on the servers is increasing rapidly so that servers will be easily overloaded for a short time, especially for a popular web server. For example the Yahoo receives 625 million views per day and the AOL Web cache system receiving 5 billion requests per day [1]. It is obvious that more and more sites have been forced to receive more unprecedented workload. That force us to find new technology of hardware and software to release the pressure the explosive growth of the Internet bring us.

In recent years the 3Com has developed the XRN technology which dramatically increased the bandwidth and interconnectivity of a network. The XRN is eXpandable Resilient Networking for short. This is a new patented LAN core technology from 3Com. The XRN allows network managers to build affordable network cores with exceptional performance and flexibility [2].

Meanwhile the Linux Virtual Server project (LVS for short) has developed software to deal with the information explosion on the Internet. Linux virtual Server is an open source project. LVS enables a computer cluster provides services as a single virtual server. It has highly scalability and highly availability [3] [4].

This paper proposes a virtual server scheme combined the new hardware technology, the XRN, with the new software package, the Linux Virtual Server, to construct a high performance and high scalability Server Cluster. Some modifications and amendments of the LVS for this combination scheme have been discussed.

The paper has been constructed as following: basic concepts of the LVS software and XRN hardware have been briefly introduced in section 2 and 3. Then in section 4 a combination

scheme of the LVS and XRN has been discussed in details, and the necessary modification and amendment to the LVS in this scheme have been proposed. In section 5 conclusions and further works for the scheme have been given.

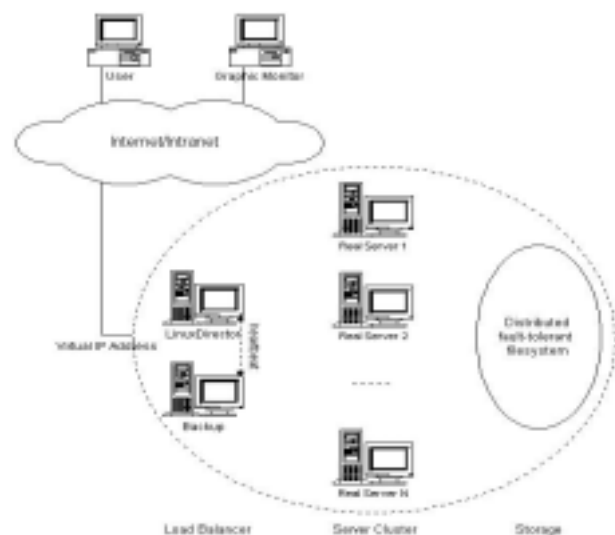
2. THE LINUX VIRTUAL SERVER

Virtual server is a highly scalable and highly available system built on a cluster of real servers. It can use inexpensive commodity hardware to achieve very high performance. The architecture of cluster is transparent to end users, and the users interact with the system as if it were only a single server. The virtual server architecture can be shown as in figure1 [3].

The real servers and distributed fault tolerant file system may be interconnected by high-speed LAN or by geographically dispersed WAN. The front-end of the real servers is a load balancer, which schedules requests to the different servers and make parallel services of the cluster to appear as a virtual service on a single IP address. Scalability is achieved by transparently adding or removing a node in the cluster. High availability is provided by detecting node or daemon failures and reconfiguring the system appropriately.[1, 3, 4, 5]

The Linux Virtual Server Project (LVS) implements layer 4 switching in the Linux Kernel. This allows TCP and UDP sessions to be load balanced between multiple real servers. Thus it provides a way to scale Internet services beyond a single host. HTTP and HTTPS traffic for the World Wide Web is probably the most common use; also it can almost be used for any service, such as FTP, SMTP, POP3, IMAP4, most TCP and UDP services.

The LVS is implemented in the Linux kernel. It is composed of 3 tiers: load balancer, server cluster and shared storage as shown in figure1.



* Supported by NSFC (Grant No.: 60173046)

Figure1. The virtual server Architecture

The LVS has employed three IP load balancing techniques, that is, virtual server via NAT, virtual server via IP tunneling and virtual server via direct routing. It adopts eight scheduling algorithms: Weighted Round-Robin, Round-Robin, Least-Connection, Weighted Least-Connection, Locality-Based Least-Connection, Locality-Based Least-Connection with Replication, Source Hashing and Destination Hashing. LVS itself runs on Linux; however it is able to load balance connections from end users, running any operating system, to real servers running any operating system. As long as the connections use TCP or UDP, the LVS can be used. [1, 3]

3. THE 3Com XRN ARCHITECTURE

The eXpandable Resilient Networking (XRN) from 3Com is a new patented LAN core technology that allows network managers to build affordable network cores with exceptional performance and flexibility. The XRN can satisfy the immediate needs of the network while won't compromise networking functionality. And when loads of the network increased, users can make use of 3Com's pay-as-you-grow option and cost-effectively upgrade to the next level. [2]

One of the key technologies of XRN is that multiple Gigabit Layer 3 switches are interconnected, behaving as a single logical multilayer switching entity called a Distributed Fabric. This Distributed Fabric provides high levels of network availability and fault tolerance to ensure continuous operation.

Another key feature is that the XRN can reduce network management costs. The XRN lets both switches in a Distributed Fabric function as a single entity - doubling the capacity of the core without an increase in administration overhead or management complexity.

In fact with the XRN, multiple interconnected Gigabit switches behave as a distributed switching fabric that grows with the network, without the physical limitations of a centralized core device. 3Com has planed to expand the XRN through 3 phases. The first phase configuration is shown as figure 2, which could be expanded as four or more switches interconnected.

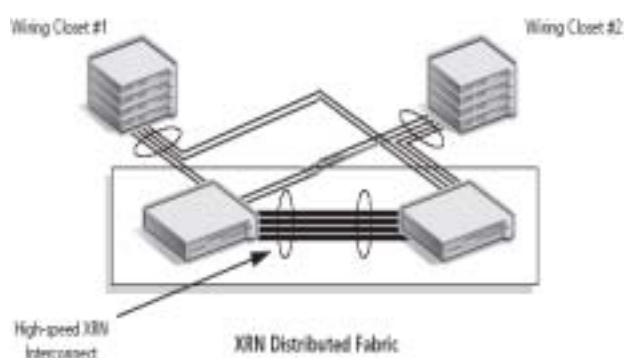


Figure2. A first phase configuration of the XRN

The XRN has three main characteristics: Distributed Device Management (DDM), Distributed Resilient Routing (DRR) and Distributed Link Aggregation (DLA). DLA guarantees high level resiliency since failure in one of the members of the Aggregated Link results in automatic redistribution of traffic

across the remaining links. XRN uses a range of high-speed interconnects to provide resiliency against external failures. DRR prevents single point of failure in the network by distributing the network's routing capability across multiple, separate switches in the Distributed Fabric [2]. Through those features the XRN can provide a high bandwidth as well as a high efficiency switch interconnection scheme.

4. A VIRTUAL SERVER SCHEME BASED ON LVS and XRN

4.1 Bottleneck of the Normal LVS models

As it has been pointed out in section 2, the LVS has employed three IP load balancing techniques: virtual server via NAT (network addresses translation), virtual server via IP tunneling (IP-IP encapsulation) and virtual server via direct routing.

In NAT model, the request packet's destination IP is rewritten to a chosen real server's IP by Load Balancer Linux Box. In this model the load balancer may be a bottleneck of whole system. Since both the request and response packets are managed by the load balancer, if the number of servers increases to around 20 or more the bottleneck problem is more severe [5]. So the scalability of a virtual server is limited.

To solve the problem, the LVS gives two alternative models that is IP tunneling and direct routing. In IP tunneling and Direct Routing model, the load balancer just schedules requests to different real servers, and the response packets can direct to the end users or follow separate network routes to the clients.

Comparing with NAT model, the load balancer can handle huge amount of requests through IP tunneling scheme; it may schedule around 100 real servers [5]. And the Direct Routing model even doesn't have the tunneling overhead.

It seems that solves the bottleneck perfectly, but all of these analyses is based on a premise that the data amount of request packets coming from end users is small comparing with that of response packets to the end users. That is right in some situations, such as web service, proxy server etc. In other occasion the data amount of request packets is equal to the amount of data of response packets, such as mail server application. Furthermore, in some occasion the incoming amount of data is great larger than output data. Taking nephogram analysis as an example, obtained output results are often a very small amount of data produced from searching and computing upon a large amount of original source data. In this situation the amount of output data almost can be ignored comparing with that of incoming data. Since the throughput of the network interface is limited, the IP tunneling and Direct Routing meet the same problem that the NAT model met. So no matter which model you chose, the Load Balancer is a bottleneck.

4.2 Full Usage of the Back up Linux Director

How to solve the bottleneck? From figure 1 it can be seen that there is a back up Linux director for fault tolerance. It implies that if the two Linux directors work together in the same time, then the throughput of the LVS will be doubled in normal case and be fault-tolerant in abnormal case. Furthermore several Linux Boxes could be used and let them working simultaneously. That is a cost efficient way to meet the challenge of huge data amount of request packets coming from end users, as well as huge data amount returned from real

servers. When loads increase, a new Linux director could be simply added to meet the increasing requests.

4.3 Necessary Modifications of the LVS

In this proposed scheme some necessary modifications of the LVS should be done. Firstly a front-end router is necessary for simply using Round-Robin manner to distribute requests to Linux Directors (Linux Boxes). Secondly all of this Linux Boxes manages the same computer cluster. As same as single Linux Box, there is some monitor software on the Linux Boxes. Some additional functions should be added into the LVS to meet the requirements. For example they should supervise state of the Linux Boxes. When one or more Linux Boxes get down, the others can take over jobs the broken one is doing; and the administrator of the LVS can take a Linux Box or more in or out of work at any time seamlessly, which won't interrupt load balance to server cluster.

4.4 Networking Requirements

In traditional IP tunneling LVS model the computer cluster comprises around 100 computers. In new scheme the number of computer can rise to 200~300 or even more. So, the computer cluster needs a robust infrastructure, and it must have high scalability, high performance and easy to administer.

The networking requirements for the load balance scheme proposed in section 4.2 are as following:

- (1) The Linux directors should be interconnected to all real servers as well as to each others.
- (2) The Linux directors should have ability to monitor incoming data flow as well as real server status.
- (3) The Linux directors should have load balance ability; the traffic load should be seamlessly and dynamically allocated between them.

All those requirements could be easily achieved by using the XRN distributed fabric or similar facilities to construct a whole computer cluster.

As what above described, the XRN offers an alternative choice for design and implementation of enterprise network cores. By using a distributed approach with multiple core products, the XRN technology allows devices to be added when and where extra performance or ports are needed. The XRN Distributed Fabric is administered as a single managed entity, with all switching and routing distributed across the multiple devices.

4.5 The New Virtual Server Scheme

The new virtual server scheme proposed would be like that: The new virtual server cluster should comprise some wiring closets, the XRN Distributed Fabric Devices and real servers. In fact, real servers are linked with wiring closets, which are interconnected to the XRN Distributed Fabric. At last, the XRN Distributed Fabric links with Load Balancer group (Linux Director Group) which is routed by a front-end router. The modified LVS package is running on the Linux Director Group.

In briefly it can be said that the new Linux Virtual Server comprises real server computer cluster, XRN Distributed Fabric and Load Balancer group with a front-end router. This new LVS scheme can be shown as figure 3.

Multi-Linux-Director scheme described above has the same functions as a single Linux Box has; moreover it solved the bottleneck problem, therefore has more reliability. Also new

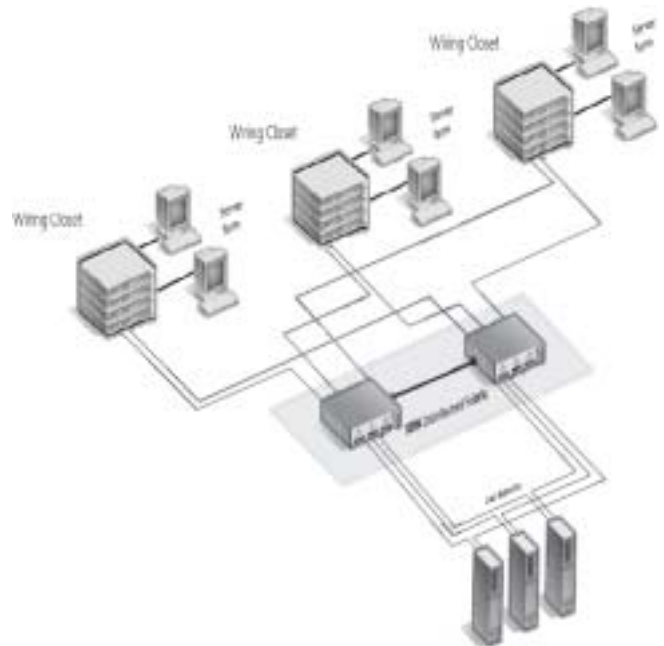


Figure 3. A Virtual Server Scheme Based on LVS and XRN

method has highly scalability and makes the best usage of computer resource.

Because using Distributed Link Aggregations technology in the XRN, the wiring closet should be dual-homed to protect against link or switch failure. The XRN Distributed Fabric behaves as a single router with multiple, active routing engines.

With the XRN and modified LVS the Load Balancer group in this scheme works simultaneously to overcome the bottleneck and avoid the single point of failure. And this scheme uses the entire Load Balancer resource, not just one as proposed in [5].

5. CONCLUSION

This paper gives a feasible plan to construct a powerful middle scale virtual server using existing hardware and software technology.

The entire architecture comprises 3 parts, that is, Load Balancer, XRN Distributed Fabric and server cluster. Each part has very high scalability and reliability. The entire architecture is robust and the Virtual Server has very high performance.

In order to support the Load Balancer group, some modification must be done in the Linux Virtual Server Package, which has been taking in one research group of the Wuhan University of Technology.

6. REFERENCES

- [1] Wensong Zhang, Linux Virtual Server: Linux Server Clusters for Scalable Network Services, Free Software Symposium 2002, October 22, 2002
- [2] Introduction to XRN: A New Direction for Enterprise Networking, 3Com Corporation, March 2002
- [3] <http://www.linuxvirtualserver.org/>
- [4] <http://www.austintek.com/LVS/LVS-HOWTO/HOWTO/>

- [5] Simon Horman, Linux Virtual Server Tutorial, VA Linux Systems Japan, K.K. March 2004.



Guo Yucheng born in 1979 is a graduate student in the school of computer science and technology, Wuhan University of technology, Wuhan China. His main research interests are in the area of distributed parallel processing, server cluster design and implementation, graphic and image processing.

An Improved BP Algorithm Based on A Variant Sigmoid Function with Three Parameters

Hu Yaogai^{1,2}, Yan Xin³, Zhang Xiaoxing⁴, Jiming Hu²

¹College of Electronic Information, Wuhan University, Wuhan, 430079, P. R. China.

²College of Chemistry and Molecular Sciences, Wuhan University, Wuhan, 430072, P. R. China.

³Department of Computer Science, Wuhan University of Technology, Wuhan, 430063, P.R. China

⁴Key Laboratory of High Voltage and Electrical New Technology of Ministry of Education, Chongqing University, Chongqing 400044, P.R. China

Email: farawayhu@sina.com Tel: +86 (27) 62040699

ABSTRACT

This paper presents a novel approach in the design of neural networks with sigmoid transfer function trained by the back-propagation algorithm. First, the artificial neural networks and back-propagation algorithm are introduced briefly. Second, a variant sigmoid function with three parameters is proposed, and then the improved BP algorithm based on it is deduced and discussed. Finally, a compared testing with the activity prediction of herbicide is given.

Keywords: Artificial neural networks, Back-propagation (BP), sigmoid transfer functions

Nomenclature

Subscript i, h, o stand for input, hidden and output, respectively; $a_i(k)$ stands for neuron k in input layer ($k=1, 2, \dots, I$, I is the sum of neurons in input layer); $a_h(k)$ stands for neuron k in hidden layer ($k=1, 2, \dots, H$, H is the sum of neurons in hidden layer); $a_o(k)$ stands for neuron k in output layer ($k=1, 2, \dots, O$, O is the sum of neurons in output layer); x_{ik} : the output value of $a_i(k)$ in input layer; X_{hk} : the total input to $a_h(k)$; x_{hk} : the output value of $a_h(k)$; X_{ok} : the total input to $a_o(k)$; x_{ok} : the computed output value of $a_o(k)$; x_{ok}^* : the required output value of $a_o(k)$; a, b, λ stand for the bias, threshold and steepness of sigmoid transfer function; w_{mn}^I : the connecting weight between $a_i(m)$ in the input layer and $a_h(n)$ in the hidden layer; w_{mn}^O : the connecting weight between $a_h(m)$ in the hidden layer and $a_o(n)$ in the output layer; x_{ok}^p stands for computed output value and of $a_o(k)$, and x_{ok}^{p*} stands for required output value of $a_o(k)$ under the pth sample; Superscript p stands for the pth training sample. ($p=1, 2, \dots, P$, P is the sum of sample set); η : learning rate; μ : momentum factor; t: iteration number.

1. INTRODUCTION

Artificial neural networks

The human nervous system consists of billions of neurons of various types and lengths relevant to their location in the body. Artificial neural networks (ANNs) are born from approach of developing intelligent systems by simulating the biological structure and the work of the human brain. The theory of ANNs is based on neurobiology, mathematics and physics.

The attractiveness of ANNs comes from the remarkable information processing characteristics of the biological system, such as nonlinearity, high parallelism, robustness, fault and failure tolerance, learning, ability to handle imprecise and fuzzy information, and their capability to generalize. As a kind of computational modeling tools, ANNs have recently emerged and found extensive acceptance in many disciplines for modeling complex problems. ANNs can be classified as a non-linear and non-parametric regression method, providing to ANNs a good flexibility that does not need a rigid mathematical model and the calibration parameters are able to be determined using data through a learning step [1, 2].

Back-propagation (BP) algorithm and its improvements

BP ANNs is one of the most important historical developments in neurocomputing. Neurons in ANNs with BP are arranged in a layered order, which is a multilayer perceptron (MLP) consisting of three different layers: 1) an input layer with nodes representing input variables to the problem, 2) an output layer with nodes representing the dependent variables, and 3) one or more hidden layers containing nodes to help capture the nonlinearity in the data. Each connection between neurons is expressed with a weight value and these connections are determined according to the training of the networks.

Standard BP learning algorithm has a unique learning principle, which is called Delta rule. The term back propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks, or the way of the error computed at the output side is propagated backward from the output layer, to the hidden layer, and finally to the input layer. Some researchers had proved that networks with biases, a sigmoid layer, and a linear output layer have the ability to approximate any function to any degree of desired accuracy, if the network contains enough "hidden" neurons between the input and output neuronal fields [3].

The main difficulty of standard BP algorithm is its slow convergence, which is a typical problem for simple gradient descent methods. And convergence to the global minimum is not guaranteed [4]. There are a number of variations on the basic algorithm that are based on other standard optimization techniques, such as conjugate gradient and Newton methods. Some novel approaches toward the regular back-propagation learning algorithm have been studied recently. As a result a large number of modifications based on heuristic arguments have been proposed to improve the performance of standard back-propagation: 1) Dynamically modifying learning rate; 2) Adjusting the steepness of the sigmoid function; 3) Improving the error function; 4) Rescaling of variables; 5) Second-order method. Although these techniques have been successful in speeding up learning for some problems, there

is no enough discussion or experiments about their abilities to avoid local minima. Moreover, they usually introduce additional parameters which are problem-sensitive.

Our work in this paper

The connecting weights among neurons and the transfer function characteristic of all of them codetermine the structure and characteristic of networks, and there are various types of alterable neurons in the body [5]. A sigmoid function is a bounded differentiable real function that defined for all real input values and that has a positive derivative everywhere. It shows a sufficient degree of smoothness and is also a suitable extension of the soft limiting nonlinearities used previously in neural networks. In this paper, we have focused attention on the sigmoid transfer function of neurons, due to the mention above and the existence of theorems which guarantee the so-called "universal approximation" property for neural networks. The paper is structured as follows. We discuss our transfer function in Section 2 and introduce a modified BP method based on it in Section 3. Results of numerical experiments are described in Section 4.

2. THREE PARAMETERS OF THE VARIANT SIGMOID FUNCTION

The transfer (activation) function is necessary to transform the weighted sum of all signals impinging onto a neuron so as to determine its firing intensity. Multilayer networks typically use sigmoid transfer functions in the hidden layers. These functions are often called "squashing" functions, since they compress an infinite input range into a finite output range. Sigmoid functions are characterized by the fact that their slope must approach zero as the input gets large. This also causes a problem when using steepest descent to train a multilayer network with sigmoid functions, since the gradient can have a very small magnitude; and therefore, cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. As mentioned above in the common BP algorithm only the weights be adjusted or changed but keep no change on transfer function of neuron during learning. But the fact is, the neurons, as the basic components compose the whole networks, with the weights among them codetermine the structure of network.

Whereas the advantage of choosing a particular transfer function over another is not yet theoretically understood [6], but the fact that the transfer function of neuron is linear or nonlinear determine the networks is linear or not indicate the key effect on whole capability of networks, and modern biology had approved the neurons have various types and lengths too [5]. Some researchers reported various success rates with different transfer functions in relation to data nonlinearity and noisiness [7]. Han, Moraga and Sinne [8] use a variant logistic function with three adjustable parameters, and each neuron is assigned a different set of values for these parameters. A bipolar sigmoid function (tanh) with asymptotic bounds at -1 and +1 is frequently used to increase the convergence speed. Other considerations have led to the use of different functions [9, 10], and even a rule-based transfer function [11] and a Morlet mother wavelet transfer function [12]. We had introduced an improved BP algorithm that link weights and Sigmoid function with two adjustable parameters [13]. In this paper, we introduce a more generalized sigmoid function showed below:

$$S_{a,b,\lambda}(x) = \frac{1}{1 + e^{-\frac{x-b}{\lambda}}} + a \quad (1)$$

Where a is the parameter for the symmetry (or bias of S) of the sigmoid function; b is the threshold value, location value or bias of x of the sigmoid function; λ is the slope (steepness), sharpness or temperature parameter of the sigmoid function.

3. THE BP ALGORITHM BASE ON VARIANTSIGMOID FUNCTION

Generally speaking, there are two different modes of training. In example-by-example training (EET), also call it as pattern mode or incremental training, the weights and other parameters of the network are updated each time an input is presented to the network. In batch training (BT) the weights and other parameters are only updated after all of the inputs and targets are presented. We discuss both types of networks in this section.

Under the p th sample, the total input to $a_h(k)$ and $a_o(k)$ are following:

$$X_{hm}^p = \sum_{k=1}^I x_{ik}^p w_{km}^{Ip}, \quad X_{on}^p = \sum_{k=1}^H x_{hk}^p w_{kn}^{Op}.$$

The sum-squared error (SSE) function of the p th sample is

$$E^p = \frac{1}{2} \sum_{k=1}^O (x_{ok}^p - x_{ok}^{p*})^2, \text{ and the SSE function of all sample set is}$$

$$E = \sum_{p=1}^P E^p = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^O (x_{ok}^p - x_{ok}^{p*})^2. \text{ Obviously, the SSE is a function of all the free parameters (including all of the } w, a, b \text{ and } \lambda \text{ in network). For sigmoidal transfer function Eq. (1):}$$

$$\frac{\partial S_{a,b,\lambda}(x)}{\partial x} = \frac{1}{\lambda} [S_{a,b,\lambda}(x) - a][1 + a - S_{a,b,\lambda}(x)];$$

$$\frac{\partial S_{a,b,\lambda}(x)}{\partial \lambda} = \frac{b-x}{\lambda^2} [S_{a,b,\lambda}(x) - a][1 + a - S_{a,b,\lambda}(x)];$$

$$\frac{\partial S_{a,b,\lambda}(x)}{\partial a} = 1;$$

$$\frac{\partial S_{a,b,\lambda}(x)}{\partial b} = -\frac{1}{\lambda} [S_{a,b,\lambda}(x) - a][1 + a - S_{a,b,\lambda}(x)].$$

The adjustment of the weights

- 1) The connecting weight w_{mn}^O between $a_h(m)$ and $a_o(n)$

In EET mode:

$$\frac{\partial E^p}{\partial w_{mn}^O} = \frac{1}{\lambda_{on}^p} (x_{on}^p - x_{on}^{p*})(x_{on}^p - a_{on}^p)(1 + a_{on}^p - x_{on}^p)x_{hm}^p$$

In BT mode:

$$\frac{\partial E}{\partial w_{mn}^O} = \frac{1}{\lambda_{on}^p} \sum_{p=1}^P (x_{on}^p - x_{on}^{p*})(x_{on}^p - a_{on}^p)(1 + a_{on}^p - x_{on}^p)x_{hm}^p \quad (2)$$

The connecting weight w_{mn}^I between $a_i(m)$ and $a_h(n)$

In EET mode:

$$\frac{\partial E^p}{\partial w_{mn}^p} = \frac{1}{\lambda_{hn}^p} (x_{hn}^p - a_{hn}^p) (1 + a_{hn}^p - x_{hn}^p) x_{im}^p \times$$

$$\left[\sum_{j=1}^O \frac{w_{nj}^{Op}}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right]$$

In BT mode:

$$\frac{\partial E}{\partial w_{mn}^p} = \frac{1}{\lambda_{hn}^p} \sum_{p=1}^P \{ (x_{hn}^p - a_{hn}^p) (1 + a_{hn}^p - x_{hn}^p) x_{im}^p \times$$

$$\sum_{j=1}^O \left[\frac{1}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) w_{nj}^O \right] \}$$

In EET mode, $\Delta w^p = -\eta_w \frac{\partial E^p}{\partial w^p}$

Consider the momentum factor:

$$\Delta w^p(t+1) = -\eta_w \frac{\partial E^p}{\partial w^p} + \mu_w \Delta w^p(t)$$

In BT mode, $\Delta w = -\eta_w \frac{\partial E}{\partial w}$

Consider the momentum factor:

$$\Delta w(t+1) = -\eta_w \frac{\partial E}{\partial w} + \mu_w \Delta w(t)$$

The adjustment of the parameters a , b and λ

1) The parameters of neuron $a_o(k)$ of output layer

In EET mode:

$$\frac{\partial E^p}{\partial a_{ok}^p} = (x_{ok}^p - x_{ok}^{p*})$$

$$\frac{\partial E^p}{\partial b_{ok}^p} = -\frac{1}{\lambda_{ok}^p} (x_{ok}^p - x_{ok}^{p*}) (x_{ok}^p - a_{ok}^p) (1 + a_{ok}^p - x_{ok}^p)$$

$$\frac{\partial E^p}{\partial \lambda_{ok}^p} = -\frac{1}{(\lambda_{ok}^p)^2} (x_{ok}^p - x_{ok}^{p*}) (x_{ok}^p - a_{ok}^p) \times$$

$$(1 + a_{ok}^p - x_{ok}^p) (x_{ok}^p - b_{ok}^p)$$

In BT mode: $\frac{\partial E}{\partial a_{ok}} = \sum_{p=1}^P (x_{ok}^p - x_{ok}^{p*})$

$$\frac{\partial E}{\partial b_{ok}} = -\frac{1}{\lambda_{ok}} \sum_{p=1}^P (x_{ok}^p - x_{ok}^{p*}) (x_{ok}^p - a_{ok}) (1 + a_{ok} - x_{ok}^p)$$

$$\frac{\partial E}{\partial \lambda_{ok}} = -\frac{1}{\lambda_{ok}^2} \times$$

$$\sum_{p=1}^P (x_{ok}^p - x_{ok}^{p*}) (x_{ok}^p - a_{ok}) (1 + a_{ok} - x_{ok}^p) (x_{ok}^p - b_{ok})$$

2) The parameters of neuron $a_h(k)$ of hidden layer

In EET mode:

$$\frac{\partial E^p}{\partial a_{hk}^p} = \sum_{j=1}^O \frac{w_{kj}^{Op}}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p)$$

$$\frac{\partial E^p}{\partial b_{hk}^p} = -\frac{1}{\lambda_{hk}^p} (x_{hk}^p - a_{hk}^p) (1 + a_{hk}^p - x_{hk}^p) \times$$

$$\left[\sum_{j=1}^O \frac{w_{kj}^{Op}}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right]$$

$$\frac{\partial E^p}{\partial \lambda_{hk}^p} = -\frac{1}{(\lambda_{hk}^p)^2} (x_{hk}^p - a_{hk}^p) (1 + a_{hk}^p - x_{hk}^p) (x_{hk}^p - b_{hk}^p) \times$$

$$\left[\sum_{j=1}^O \frac{w_{kj}^{Op}}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right]$$

In BT mode:

$$\frac{\partial E}{\partial b_{ok}} = \sum_{p=1}^P \left[\sum_{j=1}^O \frac{w_{kj}^O}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right]$$

$$\frac{\partial E}{\partial b_{hk}} = -\frac{1}{\lambda_{hk}} \sum_{p=1}^P \{ (x_{hk}^p - a_{hk}^p) (1 + a_{hk}^p - x_{hk}^p) \times$$

$$\left[\sum_{j=1}^O \frac{w_{kj}^O}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right] \}$$

$$\frac{\partial E^p}{\partial \lambda_{hk}^p} = -\frac{1}{(\lambda_{hk}^p)^2} (x_{hk}^p - a_{hk}^p) (1 + a_{hk}^p - x_{hk}^p) (x_{hk}^p - b_{hk}^p) \times$$

$$\left[\sum_{j=1}^O \frac{w_{kj}^{Op}}{\lambda_{oj}^p} (x_{oj}^p - x_{oj}^{p*}) (x_{oj}^p - a_{oj}^p) (1 + a_{oj}^p - x_{oj}^p) \right]$$

Other discussions about ANNs

A number of issues should be addressed before initiation of any network training. Some of the following issues are only relevant to BP ANNs while others are applicable to the design of all ANNs types and also used in this study.

1) Improving Generalization and the avoid to overfitting

A method for improving network generalization is to use a network that is just large enough to provide an adequate fit. If we use a small enough network, it will not have enough power to fit the data. Two other methods for improving generalization are: regularization and early stopping [14]. Generally, we can make the number of neurons of hidden layer as below:

$$H = \sqrt{I+O} + a, \quad H = \log 2^I \quad \text{or} \quad H = \sqrt{IO}.$$

Here, $a=1\sim 10$, I is the sum of neurons in input layer; O is the sum of neurons in output layer; H is the sum of neurons in hidden layer.

2) Network weight and transfer function of neurons initialization

Initialization of a network involves assigning initial values for the weights of all connections links and the transfer function parameters \mathbf{a} , \mathbf{b} , λ of all neurons. Some researchers indicate that weights initialization can have an effect on network convergence and final network architecture [15]. Some attempts have been suggested [16] for training neural networks to be insensitive to weight random variations. ASCE [17] recommends that weights and thresholds be assigned initial small random values between -0.30 and +0.30. We assigned the initializations as below: $w \sim (-0.3, 0.3)$; $a \sim (-0.2, 0.2)$; $b \sim (-0.3, 0.3)$; $\lambda \sim (0.8, 1.2)$.

3) Data preprocessing

Several preprocessing techniques are usually applied before

the data can be used for training to accelerate convergence [14]. Before training, it is often useful to scale the inputs and targets so that they always fall within a specified range through the vectors contain the minimum and maximum values of the original inputs, and the vectors contain the minimum and maximum values of the original targets. Another approach for scaling network inputs and targets is to normalize the mean and standard deviation of the training set. This procedure normalizes the inputs and targets so that they will have zero mean and unity standard deviation. The other effective procedures include the principal component analysis [18] and the genetic algorithms [19].

In our opinion, the preprocessing is not necessary but effectual for training of a high-powered network, and the trained network itself can make the preprocessing. Adopted our improved BP network, can reduce the dependent to the preprocessing. In this paper, the procedure for scaling input variables (x_i) in interval $[-1, 1]$ is:

$$\bar{x}_i = \begin{cases} \frac{2 \times x_i - (x_i^{\min} + x_i^{\max})}{x_i^{\max} - x_i^{\min}} & (x_i^{\max} \neq x_i^{\min}) \\ 0 & (x_i^{\max} = x_i^{\min}) \end{cases}$$

For scaling output variables (x_o) in interval $[0, 1]$ is:

$$\bar{x}_o = \begin{cases} \frac{x_o - x_o^{\min}}{x_o^{\max} - x_o^{\min}} & (x_o^{\max} \neq x_o^{\min}) \\ 0.5 & (x_o^{\max} = x_o^{\min}) \end{cases}$$

Where \bar{x} is the scaled value of x , and x^{\max} and x^{\min} are the maximum and minimum values of x in the database.

4) Quicken convergence and avoid false local minima

The performance of the steepest descent algorithm can be improved if we allow the learning rate and momentum factor to change during the training process. An adaptive learning rate and momentum factor will attempt to keep the learning step size as large as possible while keeping learning stable. The learning rate and momentum factor are made responsive to the complexity of the local error surface. If the new error exceeds the old error by less than a predefined ratio 1.04, the new parameters (including the weights and the parameters **a**, **b**, **λ** of transfer function) of network are accepted. This is a deterministic approximation to simulated annealing.

4. EXPERIMENTAL

In order to study the effectiveness of our BP model, we have given an application to predict the activity of herbicide by our BP networks.

The geometry of 10 herbicides were optimized by using AM1 method [20] of MOPAC2000. The Mulliken net charges of 29 atoms and the concentration (10 ppm or 100 ppm) of herbicides were selected as the 30 input values of network, and the activity of herbicides as the 4 output values of network. We divide the data into two groups: training and test subsets, and take three fourth for the training subset and one fourth for the test subset. A neural network (30-15-4) with our proposed generalized sigmoid transfer function and the usual sigmoid function ($S(x) = \frac{1}{1+e^{-x}}$) are developed to

predict the activity of herbicide, respectively. The training error curves of two networks are shown in Figures 1.

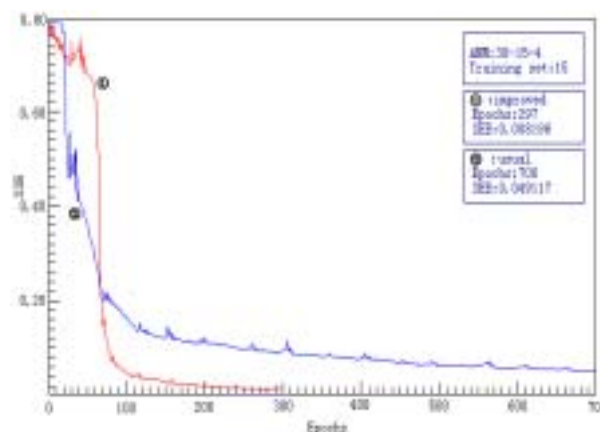


Figure1 The SSE plots for network (30-15-4) with improved sigmoid function and usual sigmoid function

From the figure we can see obviously the convergence of the network with improved generalized sigmoid function is faster than that of the network with usual one. Our proposed improved BP algorithm attains the smaller SSE in less number of times than the usual BP algorithm. The results of the prediction for the activity of herbicide are listed in Table 1.

Table 1 The results of the prediction and simulation Training subset: 1-15, Test subset: 16-20.

No.	Measured output levels				ANNs' output levels			
	Cole		Barnyard grass		Cole		Barnyard grass	
	root	stem	root	stem	root	stem	root	stem
1	0.812	0.744	0.878	0.780	0.812834	0.747465	0.877915	0.784711
2	0.800	0.737	0.850	0.765	0.800233	0.743131	0.850149	0.762507
3	0.971	0.944	0.963	0.708	0.973356	0.940973	0.960317	0.708584
4	0.952	0.731	0.900	0.542	0.951350	0.743907	0.899929	0.547906
5	0.948	0.761	0.905	0.763	0.948333	0.759716	0.905120	0.762638
6	0.975	0.920	0.930	0.805	0.973004	0.922613	0.930920	0.803159
7	0.970	0.880	0.920	0.783	0.972818	0.879810	0.919617	0.784846
8	0.970	0.760	0.850	0.736	0.773123	0.742989	0.850247	0.736560

9	0.837	0.776	0.900	0.763	0.836710	0.775151	0.899632	0.763409
10	0.920	0.791	0.925	0.777	0.919279	0.791662	0.924717	0.776366
11	0.797	0.746	0.833	0.763	0.795183	0.742592	0.836320	0.762321
12	0.970	0.940	0.958	0.833	0.965628	0.942154	0.956981	0.826620
13	0.958	0.925	0.932	0.805	0.958968	0.923164	0.932561	0.801196
14	0.905	0.825	0.900	0.777	0.906402	0.825172	0.899520	0.781424
15	0.972	0.880	0.950	0.819	0.969737	0.879905	0.949808	0.819140
16	0.966	0.850	0.955	0.792	0.955356	0.786168	0.946855	0.763137
17	0.824	0.790	0.860	0.750	0.781852	0.750177	0.875145	0.780276
18	0.796	0.745	0.883	0.750	0.776525	0.742497	0.838409	0.729940
19	0.948	0.925	0.933	0.805	0.952103	0.863225	0.942489	0.829903
20	0.945	0.791	0.900	0.721	0.969618	0.769427	0.937891	0.766141

The SSE of all 20 samples is 0.293357, the SSE of 15 training samples is 0.00196, and the SSE of 5 test samples is 0.285161. Simulation and prediction results with our BP networks are satisfactory.

5. CONCLUSIONS

In this paper we developed a modified algorithm based on three parameters of the variant sigmoid function that have more powerful nonlinear mapping capability than the classical one. The results presented show that the proposed algorithm is effectual. Compared with the usual BP networks, the efficiency of our improved BP networks shows significant improvements and the classification performance of it is also satisfactory. The proposed generalized sigmoid transfer function will be a promising alternative to the classical sigmoid function.

6. REFERENCES

- [1] F. Despagne, D.L. Massart, "Neural networks in multivariate calibration", *Analyst*, 123, 1998, pp.157~178.
- [2] J. Raimundo, R. Narayanaswamy, "Simultaneous determination of relative humidity and ammonia in air employing an optical fibre sensor and artificial neural network", *Sens. Actuators B* 74, 2001, pp.60~68.
- [3] K.I. Funahashi, "On the approximate realization of continuous mappings by neural networks", *Neural Networks*, Vol.2, No.3, 1989, pp.183~192.
- [4] D.H.Rouvray, "Making the right connection", *Chem. Br.* 29, 1993, pp.495~498.
- [5] R.J.Schalkoff, *Artificial Neural Networks*, New York: McGraw-Hill, 1997.
- [6] M.H.Hassoun, *Fundamentals of Artificial Neural Networks*. MIT Press, Cambridge, MA, 1995.
- [7] J. Moody, N.Yarvin, *Networks with learned unit response functions*. In: J. Moody, et al. (Eds.). *Advances in Neural Information Processing Systems*, Vol. 4. Morgan Kaufmann, San Mateo, CA, 1992, pp. 1048~1055.
- [8] J. Han, C. Moraga et al, "Optimization of feedforward neural networks", *Eng. Appl. Artif. Intell.*, Vol.9, No.2, 1996, pp.109~119.
- [9] P.A. Shoemaker, M.J. Carlin et al, "Back-propagation learning with trinary quantization of weight updates", *Neural Networks*, 4, 1991, pp.231~241.
- [10] H. Aapo, "An alternative approach to infomax and independent component analysis", *Neurocomputing*, 44~46, 2002, pp.1089~1097.
- [11] Zhang J.W., M.Ferch, "Extraction and transfer of fuzzy control rules for sensor-based robotic operations", *Fuzzy Sets and Systems*, 134, 2003, pp.147~167.
- [12] Zhang X. Y., Qi J. H. et al, "Prediction of programmed-temperature retention values of naphthas by wavelet neural networks", *Computers and Chemistry*, 25, 2001, pp.125~133.
- [13] Hu Y. G., Li K. Y. et al, "An Improved BP Algorithm of Neural Networks", *Journal of Wuhan University (Natural Science Edition)*, Vol.45, No.1, 1999, pp.25~29.
- [14] I.A. Basheer, M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and Application", *Journal of Microbiological Methods*, 43, 2000, pp.3~31.
- [15] W. Schmidt, S. Raudys et al, "Initialization, backpropagation and generalization of feed-forward classifiers", In: *Proceeding of the IEEE International Conference on Neural Networks*, 1993, pp.598~604.
- [16] M. Conti, S. Orcioni and C. Turchetti, "Training neural networks to be insensitive to weight random variations", *Neural Networks*, 13, 2000, pp.125~132.
- [17] ASCE, "Artificial neural networks in hydrology. I. Preliminary concepts", *J. Hydro. Eng. ASCE* 5, 2000, pp.115~123.
- [18] Chen D.Z., Chen Y.Q. and Hu S.X., "A pattern classification procedure integrating the multivariate statistical analysis with neural networks", *Computers Chem.*, Vol.21, No.2, 1997, pp.109~113.
- [19] Jasmina Arifovic, & Ramazan Gencay, "Using genetic algorithms to select architecture of a feedforward artificial neural network", *Physica A*, 289, 2001, pp.574~594.
- [20] M.J.S. Dewar, E.G. Zebisch et al, *J. Am. Chem. Soc.* 107, 1985, pp.3902.

Implementing Synchronous Multicasting in Switch-Based Cluster Systems

Feng Ping, Lei Yanjing, Liu Junrui
 College Of Computer, Northwestern Polytechnical University
 Xi'an Shaanxi 710072 China
 Email: fengping_01@163.com

ABSTRACT

A new realtime-based wormhole routing technique is presented. This kind of routing method that can implement synchronous multicasting in switch-based cluster systems has high-performance and low cost. By using this technique, we have developed a kind of computer network, which consists of an RTnet network interface card (RTNIC), RTnet network switch and commutation protocols. The RTnet is designed to provide a low-latency and high throughput end-to-end interconnects. The performance of Multicasting and realtime is better than the Myrinet. The build-up time of channel is shorter than that of SCI (Scalable Coherence Interface) and FC (Fiber channel), the throughputs higher than that of SCI and FC. By testing, the establish time of connection can be less than $0.1 \mu s$; The 10 bits data can be transferred through the crosspoint in 8ns; The end to end latency is lower than $1 \mu s$.

Keywords: realtime-based wormhole routing, synchronous Multicasting, low-latency, high- throughput, end-to-end interconnects

1. INTRODUCTION

Current the main routing technique of cluster is wormhole routing. In wormhole routing, there are two kind of realization technique. One is blocking wormhole routing and another is buffered wormhole routing including Virtual Channels.

Blocking wormhole routing has the characteristic of low transmit latency and low cost [1]. The hardware delay between any two nodes is almost independent of the path length. Each node uses a flit buffer to hold one flit, at a time. All data flits in the same packet follow the same path that the header traverses. If there is no conflict, successive packets are forwarded in a pipelined fashion. Except the flit of header, others of flits do not include the routing message. The sequence of flits must hold the link channel successive, which cannot be interrupted by other flits of packed. This problem which blocking wormhole routing has will not allow broadcasting and multicasting. The blocking wormhole routing will also cause deadlock situation easily.

Buffered wormhole routing allows broadcasting and multicasting. Craig B. Stunkel put forward a multicast wormhole [2], which has a central queue. In the central queue architectures, the blocked flits are buffered in a central queue, which provides a dynamically shared resource for input and output ports. If the load among the input ports is unbalanced, the central-buffer-based scheme is likely to benefit because

the ports have shared access to a larger buffer and can potentially use nearly the entire space available even when some or all of the intended output ports are busy with other traffic. There is no guarantee that a packet arriving at a switch will find enough space in the central buffer to be completely stored. If the central buffer cannot store the entire blocked packet, that will not guarantee to prevent multicast deadlock. To address this problem, multicast wormhole made minor modifications to the basic central buffer free-space logic that are similar to virtual cut-through (VCT) operation. Wormhole flow-control designs can be augmented to provide the aspects of VCT that are essential for multicasting, given sufficient buffering capability within each switch. To use a central buffer for emulating VCT, the total central buffer size must be as large as or larger than the largest packet to be buffered.

VCT designs perform flow-operation is required to avoid the dependence between output ports which can lead to multicast deadlock. Virtual cut-through is to allow each physical channel to emulate several virtual channels, and to construct a virtual network in which the worms cannot form cycles. The virtual channels share the physical wire (or wires) provided by the physical channel, but the switch maintains a separate queue for each virtual channel. This solution has also been implemented in hardware. For example, each physical channel of the i Warp machine supports 4 virtual channels, and each physical channel of the J-Machine supports 2 virtual channels [3].

The multicast wormhole made some improvements on multicast, but which exits some problem. The one problem is that larger buffer is needed and some complex approach needs to be solved. Another problem is that time spending of asynchronous replication and the central buffer is larger. The virtual channel performs flow-control on a packet basis. The size of the packet is larger, the transmission time of packet is longer and the buffer is bigger. If the size of the packet is smaller, the throughput is smaller.

In allusion to stating above, and the need of teal time systems, we present a realtime-based wormhole routing technique. Realtime-based wormhole routing technique bases on the wormhole routing technique. The header flit includes the routing message of destination node. Each node will save the routing message until the trailing packet flits passing. Realtime-based routing technique overlaps the transmission of successive packets by using the network resources along the selected path as a transmission pipeline. The packet bypasses an intermediate node without copy and storage delays. The data flits in one packet can be interrupted and be saved in the buffer temporarily, when another packet needs to pass through the node which holding by the flit.

For real-time need, the packets have priority. The flits of high priority packed can suspend transmission of low priority flits temporarily. The blocking flits of low priority can be saved in message buffer. Because of the synchronous replication, the performance of real time is better than wormhole routing,

For validating this theory, we successfully developed an implementation prototype---RTnet, which includes PCI network interface card (RTNIC), 16-port network switch and network protocol software.

RTnet network is based on the real time wormhole routing technique. When RTnet network interface card (RTNIC) has a request if the output port of a switch is free, the controller of network switch will establish channel by closing appropriate crosspoint in the RTnet switch. After the channel established, the package of source node can be directly delivered to the destination node without message buffer. If the channel is not free, the flits can be directly blocked to message buffer of different place.

The cluster system is applied for high speed interconnects. Because of a short distance between nodes, nodes limited in number, a unitary environment, and especially a very low error rate reaching 10^{-12} , the communication protocols are simplified for reducing the communication latency between protocol layers. Compare with Myrinet [4] and Fiber Channel [5], this kind of routing method has a simply physical layer protocol. The RTnet defines the protocols of physical layer, link layer and network layer. RTnet provides three kind necessary services. The class 1 is a service that provides dedicated connection; the class 2 Frames defines that this service allows one N-port to transmit consecutive frames to multiple destinations without establishing a dedicated connection with any specific N-port; the class 3 Frames defines the real time broadcasting and multicasting. RTnet has the merits of scalability, low latency and high-bandwidth for a lowly parallel system. RTnet can satisfy the need of real time computing.

The paper is organized as follow. Section 2 is the Gigabit RTnet architecture; Section 3 is the RTNIC architecture; Section 4 analyses the transmission time of real-time wormhole routing technique, which RTnet uses. Section 5 analyses the priority routing, which RTnet uses. Section 6 analyses the throughput of RTnet. Section 7 shows performance evaluation results. Section 8 is a conclusion.

2. THE GIGABIT RTNET ARCHITECTURE

The block diagram of Gigabit RTnet architecture is as shown in Figure 1. 16 node-computers are connected to the RTnet switch. Each node can request to establish communication channel with any other node. As long as the destination ports free, the channel will be successfully established immediately. The RTnet switch is built with the crosspoint, switch controller, message buffer, Full-Duplex SerDes transceiver and optical transceiver.

The crosspoint adopt a kind new chip M21110 of Mindspeed Co [6]. The M21110 has seventeen IO ports. Each port has bandwidth as high as 3.2Gbps. The M21110 has two components, as shown in Fig. 2.

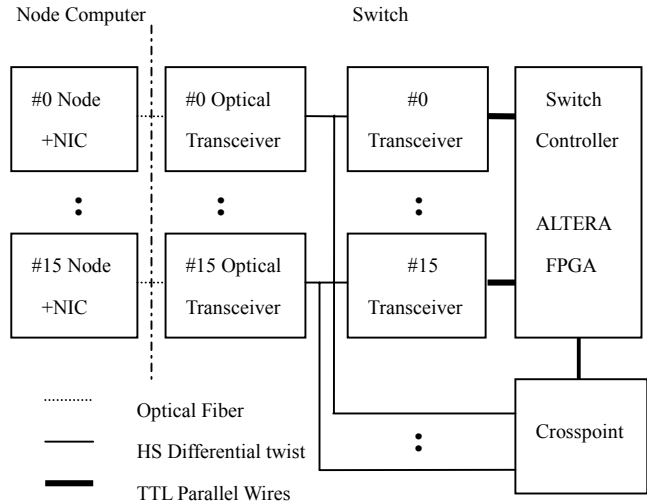


Figure 1 RTnet Architecture

The lower one is a set of 17×17 differential crosspoints, and the upper one is switch setting circuit, which control the appropriate crosspoint to be closed or opened. The M21110 chip has only one input. In order to share the unique input, we designed a large scale FPGA of ALTERA Co for controlling the switch setting circuit, as shown in Fig. 1.

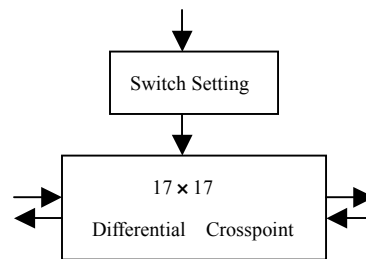


Figure 2 RTnet Switch

Due to the speed limitation, the ALTERA FPGA chip cannot work at more than 1Gbps. For each channel, we use CX27207 as Transceiver to deserialize the serialized input into parallel code at 125 Mbps, and to analyze the RTnet commands.

According to the RTnet command, the ALTERA FPGA will control M21110 to establish appropriate communication channel. In the same way, the channel is turned off according to the canceling command; Each RTnet switch has two kinds of ports. One is N-port, which connect with PNIC of host. Another is E-port, which connect with an RTnet switch. Those kinds port can be set by initialization.

3. RTNIC CONSTRUCTION

The RTnet network interface card (RTNIC) is a standard PCI card, which includes the PCI bus interface logic, data splitting logic, data synchronous logic, Serializer/ deserializer (SerDes) logic, receive and transfer FIFO, and optical transceiver. PCI bus interface logic and communications controller is built with FPGA of ALTERA Co. Transceiver is M27207 of Mindspeed Co [6]. M27207 has four Full-Duplex transmitters and receivers, also has clock recovery and synthesis circuits, 8B/10B encoders and decoders. RTNIC architecture is as shown in Figure 3.

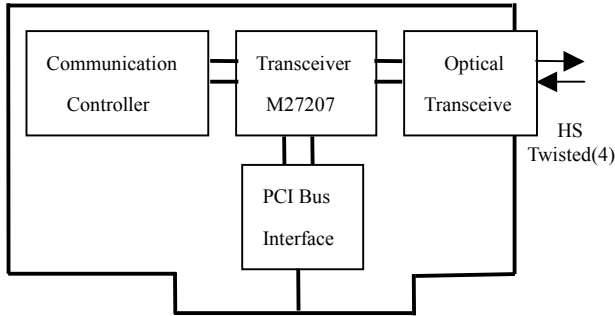


Figure 3 RTnet network interface card

For communication between host and RTNIC, The RTnet adopts VIA (virtual interface architecture).

The RTNIC has independently four receivers, transfers and buffers, which can support parallel communication mode and realize four parallel communications between RTNIC and switch, also between RTNIC and RTNIC. The RTNIC can operate at bit rates as low as 1 Gbps per channel and provide up to 12.75 Gbps of duplex raw data bandwidth

4. THE TRANSMISSION TIME ANALYSIS

Comparing with wormhole routing, realtime-based wormhole routing has some more flexibility method and lower latency. In blocked situation, if we do not consider the priority, the transmission time of RTnet routing is the same as wormhole routing for the same packages.

In routing, if B is the width of channel, L is the length of package, and a package consists of m flit. D is the path length between the source and destination node. Before the blocking released, there are n ($n < m$) flit into some middle nodes. If total blocking time is T_B , the length of flit is L_F . When having relieved from blocking, the transmission time of remnant flit is stated below:

$$T_w = \frac{L_F}{B} \times D + T_B + \frac{(m - n)}{m} \times \frac{L}{B} \quad (1)$$

In wormhole routing, the flit (L_F) is small, universality, $L_F \ll L$,

so m is larger. From formula (1), we know, if reducing the T_w , the n must be larger.

If n is the number of node passed by the header, p is the number of node passed by the tail. When the packages are blocked, the nodes amount, which is shared by other node in the route, is stated as below:

$$N = D - (n - p) \quad n \leq D, p \leq D - 1 \quad (2)$$

If we consider the channel shared, the transmission time will multiply by the coefficient of shared channel, which is less than 1.

5. THE REALTIME-BASED ROUTING ANALYSIS

Switch-based parallel systems fall into two broad classes:

- (i) Systems based on regular interconnect and
- (ii) Systems based on irregular interconnect.

In irregular topologies, deadlock free routing of messages in such networks is a challenging problem. The schemes and designs of RTnet are applicable to all categories of switch-based parallel systems.

In strict or pure wormhole routing, switches buffer only a single flit of an arriving worm and cannot accept the next flit until this buffer is freed. For a multidestination worm that has to be forwarded to multiple output ports, this buffer cannot be freed until the switch can forward the flit to all (required) output ports. Thus deadlock can easily occur if two multidestination packets that must be forwarded to the same set of output ports arrive at a switch with one of the worms reserving a subset of the output ports and the other worm reserving the rest. This deadlock problem exists even if the switch can buffer more than one flit.

There are two kind of method to solve deadlock problem, one is asynchronous replication and another is synchronous replication. In asynchronous replication, the blocked branches don't block other branches. For irregular topologies, routing of asynchronous replication can be performed much like the routing described for bidi-MINs by assuming a tree structure superimposed on the irregular network. Such tree structures are typically used in irregular networks to provide deadlock-free routing of unicast messages. In synchronous replication, the costly feedback architecture is required.

We present a new synchronous replication approach. A bit-string encoding scheme is designed to implement multidestination worms. This encoding allows a multicast/broadcast to be implemented using a single communication phase.

The function of synchronous broadcasting and multicasting are realized by hardware. According to distinction node address of command, the RTnet switch will establish channels by closing corresponding crosspoints. Within a switch, the

RTnet allow to interrupt the current connection, and to broadcasting or multicasting at the same time. After the channel established, the package of source node can be directly delivered to the corresponding destination nodes. The RTnet switch also provides package transmission with priority function of hardware. The transmission of high priority RTnet commands is realized by interrupting current connection. It is shown that multidestination worms can be implemented on these switch architectures with very little additional hardware.

In each physical channel of a node, there are four queues. The flits in the queue has some priority, the high priority flits can transmission first. The destination address saved in the header of the queue, which does not be deleted until receiving the trailing of a packet. If there is the same priority, the flits can be transmission alternate like VTC. There are three kinds of advantage that the RTnet performs flow-control on flits basis. The first is easy to realize the synchronous replication; the second is to provide deadlock-free routing; the third is of high avail throughput, because the transmission of flits need not carry the destination address.

The high priority is set for reaving, broadcasting and multicasting. Broadcasting and multicasting are common operation in parallel programs. For real time application, it would be beneficial to be able to reduce the latency of this operation as much as possible. As a full interconnect crosspoint, the RTnet can implement end-to-end, real-time multicast and broadcast easily, when the broadcasting and multicasting have some high priority.

Multicast traffic evaluation raises some issues beyond unicast traffic. The definition of multicast latency can be defined as the latency of the last received message of the multicast.

Condition in no reaving, broadcasting and multicasting, T_B is the transmit time broadcasting or multicasting within a switch. T_M is the transmit latency of point to point.

$$T_B < 2T_F + T_M \quad (3)$$

In a parallel system the diameter of multistage network commonly would be equal or lower than two, T_L is the transmit latency of end to end. The maximal transmit time of multistage network T_{Dmax} is:

$$T_{Dmax} = 3(2T_F + T_M) + T_L \quad (4)$$

For unreal time application, broadcasting and multicasting adopt message-duplicated arithmetic. We propose a method for constructing an optional multicast tree to reduce the latency of broadcasting and multicasting.

The RTnet solves the clock synchronization problem by using the function of realtime-based broadcasting and multicasting.

6. THE THROUGHPUT ANALYSIS

We must calculate the total number of packets crossing each

link to find out the maximum throughput per node during all-to-all communication. We have narrowed down the scope of RTnet topologies. Using 16-port switches, each switch is connected to another switch with one link, forming a linear array of switches. This is a minimally connected network, which is the least expensive method of connection, but provides poor bandwidth between switches.

We assume a fairly symmetric network, where all switches are connected to the same number of nodes and switches.

Let N be the total number of nodes and S be the total number of switches. For each switch, let j be the number of ports connected to switches and let k be the number of ports connected to nodes.

The total number of packed that must flow through a single link is called the *hot-link*. A general form for the total number of packets passing through each switch-to-switch link is

$$hot-link = \frac{N(N-k)(h_{sw-sw})}{j.S} \quad (5)$$

Where h_{sw-sw} is the average number of hops between switches. For lowly parallel system, the average hops between switches is 1 h_{sw-sw} 3.

We assume that the RTnet packed cumulative overhead for any size message; the number of bytes is 16. Based on the values, the efficiency is calculated as

$$Efficiency = \left(\frac{M}{M+16} \right) \quad (6)$$

Where M is the message size in bytes.

$$B_{eff} = B_{gross} \times Efficiency \quad (7)$$

B_{eff} is the available bandwidths. The least gross bandwidths B_{gross} is 1Gbps. The available amount of switch-to-switch bandwidth is found as the number of packets sent by a node that must travel switch-to-switch, divided by the hot-link, multiplied by the effective bandwidth B_{eff} .

$$B_{sw-sw} = \left(\frac{N-k}{hot-link} \right) B_{eff} \quad (8)$$

Notice that Eqs. 5 and 8 are only counting the $N-k$ packets that are destined for another switch. The remaining $k-1$ packets to the nodes on the same switch still have full link bandwidth from source to destination as given below in Eq. 9.

$$B_{sw} = \left(\frac{k-1}{hot-link} \right) B_{eff} \quad (9)$$

Thus, the overall available bandwidth per node is

$$B_{\text{avail}} = \left(\frac{(N - K)B_{\text{SW-SW}} + (k - 1)B_{\text{SW}}}{N - 1} \right) \quad (10)$$

7. THE PERFORMANCE EVALUATION RESULTS

The graphs of figure 4 displays performance measures, the transmission latency reckoned in cycle is on the y-axis and flits number of a packet is on the x-axis. Each output (input) port can send (receive) 1 byte every 8 nsec cycle. We count latency in cycles and we assume size of a flit is from 2 byte to 200 byte. By testing, the setup time of one channel is 7 cycles; the end-to-end latency needs 38 cycles.

8. CONCLUSION

The RTnet has a full interconnect construction; the each port can realize a commutation of end-to-end and synchronous broadcast.

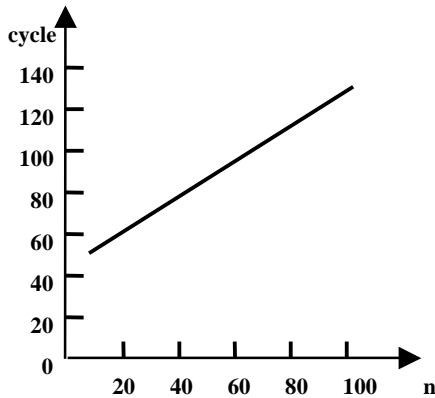


Figure 4 performance measures

The RTnet routes the destination node by the address of destination. The routing commands are very short. The build-up time of channel is shorter than that of SCI and FC, the throughputs higher than that of SCI and FC. By testing, the establish time of connection can be less than $0.1 \mu s$; The 10 bits data can be transferred through the crosspoint in 8ns; The end to end latency is lower than $1 \mu s$. Because of the simple construction and low transmission latency, system rebuilding can be realized flexibility. We think that the RTnet can satisfy the need of high performance and realtime computing for a cluster system.

9. REFERENCES

- [1] Ni L. A Survey of Wormhole Routing Techniques in Direct Networks. Computer, 1993, 2(1):62~76
- [2] Craig B.Stunkel, "Implementing Multidestination Worms in switch-Based Parallel Systems: Architectural

Alternatives and their Impact", Proceedings of the 24th annual international symposium on Computer architecture, June 1997, Denver, Colorado, United States pp 50-61.

- [3] Richard J. "On the Benefit of Supporting Virtual Channels in Wormhole Routers", Proceedings of the eighth annual ACM symposium on Parallel algorithms and architectures June 1996, Padua, Italy
- [4] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W. -K. Su, "Myrinet: A Gigabit-per-Second Local Area Network", IEEE Micro, Feb. 1995, pp. 29-36.
- [5] E. M. Frymoyer, "Fiber Channel Fusion: Low Latency, High Speed", Data Communications, <http://www.data.com>, Feb, 1995.
- [6] Mindspeed Co. "Mindspeed Technologies", <http://www.Mindspeed.com>, 2002.

Feng Ping is an adjunct professor of college of computer, in Northwestern Polytechnical University, Xi'an Shaanxi P.R.china. Her research interests are in the system Architecture of computer network, parallel processing and grid computing.

A Study of Personalized Network Based on Multi-Layer Packet Classification and ACL*

Cheng Chuanhui, Li Layuan, Xiang Yang
Wuhan University of Technology

Email: sammic@hotmail.com Tel.: 13307113650

ABSTRACT

In the future Internet, the personalized network service will be the certain trend. This paper describes how the network supports personalized network, and according to Galnet-3 device, introduces ACL and packet classification implementation mechanism. An example is given in this paper. Finally, the future development of personalized network is talked about.

Keywords: personalized network, ACL, packet classification, multi-layer switch

1. INTRODUCTION

While the competition of the network service provider is more and more fierce, the key is not only the bandwidth and access speed, but also the innovation of service property and operation property. Whether it could be understood that the customers' action and providing the customized service is the crucial factor to judge a network service provider holding key competition or not.

The network service provider must take it important to distinguish customer action. Only to differentiate customers and supply the different, scalar and personalized service, can a provider keep old customers and attract new customers. To supply personalized services effectively, for a network provider, it is very important to select a proper network technology and design layout network layer and application layer to support personalized service.

Personalized network service consists of network service and information service. Network service is to supply different scale network access service, network security, network performance and network management. Personalized information service is to create and manage the interest information. The customer can get information by his own interest.[1]

This paper mainly proposes how to realize a module of personalized network service, which meets customer's need via multi-layer packet classification and access control list technology in the network device. This paper is structured as follows. Related background knowledge such as layer-2, layer-3 switch will be introduced in section 2. Layer-4 to layer-7 switch, which is potential support, personalized network service is also introduced in this section. Section 3 proposed ACL (Access Control List) of multi-layer switch. Then the implementation of multi-layer packet classification is introduced in section 4. Section 5 gives some examples of

personalized network implementation via multi-layer classification and access control list. Section 6 concludes this paper and some further research directions are put forward.

2. BACKGROUND OF PERSONALIZED NETWORK SERVICE

The layer, which supports personalized network service, includes network layer and system layer. As personalized network service must lie near the customer, Ethernet, which is the mainstream of access network technology, is the preferred network layer.

OSI is a layered network design framework, which establishes a standard so that devices from different vendors can work together. Network addresses are based on this OSI Model and are hierarchical. The more details that are included, the more specific the address becomes and the easier it is to find.

The Layer at which the switch operates is determined by how much addressing detail the switch reads as data passes through.

Switches can also be considered as low end or high end. A low-end switch operates in Layer 2 of the OSI Model and can also operate in a combination of Layers 2 and 3. High-end switches operate in Layer 3, Layer 4, or a combination of the two.

2.1 Layer-2 switch

Layer 2 switches operate using physical network addresses. Physical addresses, also known as link-layer, hardware, or MAC-layer addresses, identify individual devices. Most hardware devices are permanently assigned this number during the manufacturing process.

Switches operating at Layer-2 are very fast because they're just sorting physical addresses, but they usually aren't very smart—that is, they don't look at the data packet very closely to learn anything more about where it's headed.

2.2 Layer 3 Switches (The Network Layer)

Layer 3 switches use network or IP addresses that identify locations on the network. They read network addresses more closely than Layer 2 switches—they identify network locations as well as the physical device. A location can be a LAN workstation, a location in a computer's memory, or even a different packet of data traveling through a network.

Switches operating at Layer 3 are smarter than Layer 2 devices and incorporate routing functions to calculate actively the best way to send a packet to its destination. But

*The work is supported by National Natural Science Foundation of China under grant number of 60172035 & 90304018 and key project of Wuhan.

although they're smarter, they may not be as fast as the former if their algorithms, fabric, and processor don't support high speeds.

2.3 Layer 4 Switches (The Transport Layer)

Layer 4 of the OSI Model coordinates communications between systems. Layer 4 switches are capable of identifying which application protocols (HTTP, SMTP, FTP, and so forth) are included with each packet, and they use this information to hand off the packet to the appropriate higher-layer software. Layer 4 switches make packet-forwarding decisions based not only on the MAC address and IP address, but also on the application to which a packet belongs.

Because Layer 4 devices enable you to establish priorities for network traffic based on application, you can assign a high priority to packets belonging to vital in-house applications with different forwarding rules for low-priority packets such as generic HTTP-based Internet traffic.

Layer 4 switches also provide an effective wire-speed security shield for your network because any company- or industry-specific protocols can be confined to only authorized switched ports or users. This security feature is often reinforced with traffic filtering and forwarding features.

In a word, to support personalized application, the layer-2 or layer-3 is not suitable. Layer-4 to Layer-7 switch must be applied to avoid content scotoma. By multi-layer switch, it is easy to distinguish which application a packet belongs to and to give it a policy.

3. ACCESS CONTROL LIST

ACL(access control list) can decide whether forwarding a packet by configuring access control matrix.

Taking an example of Galnet-3 device which is produced by Marvel corporation^[2], each Non-multicast IP Address can belong to one of 32 address groups referred to "Access Groups". An Access Group may be an IP end-station, server, group of servers, network, group of networks etc. The Access Group for each IP address is specified in the IP Route_Entry<Access_Group> field. The Galnet-3 device maintains an Access Matrix of 32 rows by 32 columns, which enables control from one Access Group to another.

Each entry in the Access Matrix represents a 2-bit policy as follows:

- 00 - Route
- 01 - Drop
- 10 - Intervention: Trap and send to CPU
- 11 - up to Flow Classification Unit

After completing all routing functions on the packet, it should be sent to the Flow Classification for further treatment, which we will describe in section 4. This feature adds another layer of security by defining a policy on a Source-Destination pair basis, in addition to the Source or Destination policy found in IP_Route_Entry<DA_CMD>/<SA_CMD> and the Per Flow policy.

The Access Matrix can be enabled by setting IP_Route_Control_Reg<Access_Matrix_En>=1. The CPU should initialize the Matrix prior to this. It must not rely on the defaults found in the Matrix after power-up.

4. MULTI-LAYER PACKET CLASSIFICATION

4.1 Two-stage packet classification

For personalized network, layer-4 switching^[3] has been proposed. Routing and QoS lookups are integrated into a single framework to fulfill layer-4 switching, and therefore, the forwarding database of a router consists of a large number of filters to be applied on multiple header fields. The deployment of a large-scale packet filtering mechanism [4] makes it feasible to implement layer-4 or 5 switching at edge routers or at the front-end of server farms.

To support layer-4 service differentiation and fulfill the personalized network, a two-stage packet classification mechanism is applied. Routing decisions must be made at the input port, but most of service differentiation — buffer management and packet scheduling—is performed at the output port. There is a large difference between the search spaces of routing lookup and QoS lookup. The size of routing table is very large and ever increasing with the growth of Internet, but the filtering rule set of QoS classification is small and remains stable.

Conventional routing lookup is based solely on destination addresses, which is a one-dimensional search, but QoS lookup is based on multiple fields, which is a multidimensional.

Figure 1 shows the two-stage packet classification procedure. Input packets first match the forwarding table for layer-2 or layer-3 switch, then enter the switching fabric. For personalized network service, the packets enter the QoS Table for packet classification and classed to different flow. For different flow, different policy will be applied; hence the differential and personalized service is supplied. In the procedure, forwarding can be redirected by TCP/UDP port other than the forwarding table by mac/IP.

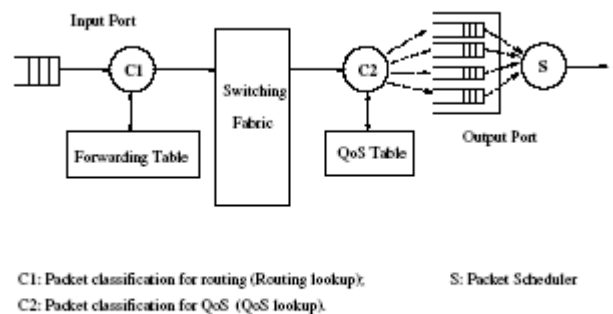


Figure 1: The architecture of the two-stage packet classifier at core routers

4.2 Implementation of packet classification

In the Galnet-3 device, multi-layer flow classification may be triggered to override Class-Of-Service and routing/switching decisions for a given packet with sophisticated

criteria and also to provide extended filtering capabilities. This function may be used as an extension to the IPv4 routing services and also as an extension to the regular Layer 2 switching process.

The implements flow classification per packet according to one or more bytes in the packet, for Galnet-3 device, any 16 individual bytes pattern/mask pair, where each byte offset exists within the first 64 bytes of the packet, and also according to the receive port and/or the Vlan-ID. Such is the typical Layer 3/4 IP fields: IP Protocol, IP Source, IP Destination, UDP/TCP source port, and UDP/TCP destination port. This is typically used to classify application flows.

A basic classification can also be performed based on the IPv4 Protocol field or TCP/UDP port numbers. A policy can be applied per flow, where each flow can be dropped, passed to the CPU, passed for further treatment, or only classified and enqueued back to the transmission queue(s). This can be used for implementing application-based access control list.

The classification may also modify the target port and device set by the Layer 3 routing or Layer 2 switching for "Layer 4 switching" and load balancing over link aggregate (trunk). A packet passes to the Flow Classification Unit based on one of three criteria: MAC information, Receive port, or IPv4/IPX routing information.

There are four master flow classification functions that support multi-protocol classification. Each flow classification function is defined by a flow type; of the four flow types, one can be used for IPv4 and one for IPX.

The Flow Classification Table can maintain up to 128K exact-match flows. Unmatched flows can either be sent to the central CPU.

5. AN EXAMPLE

An example will be listed to show how to construct a personalized network via ACL and packet classification technology.

Personalized example:

To manage a switch/router, telnet ip, but the Solution: because the ordinary customer is in some vlans which we know the VID. So we can implement the packet classification rule as (VID,DIP,Layer4 port).VID matches the ordinary vlan id, DIP matches the device ip,and Layer4 port match telnet service known port , And the action is "drop this packet". So the ordinary customer cannot telnet device.

6. CONCLUSION

In the future Internet, the personalized network service will be the certain trend. This paper describes how the network support personalized network, and according to Galnet-3 device, introduces ACL and packet classification implementation mechanism. An example is given. With the development of technology and the practical application, personalized network will make progress.

7. REFERENCES

- [1] Cai Ming, "Technology Research on Personalized network Service", Telecommunication Science,2003(7)
- [2] Galileo Inc.GT48510/A Converged Voice/Data Network Switch Processor User Manual Revision.1.0[Z],2000
- [3] Haining Wang , Kang G. Shin . " Layer-4 Service Differentiation and Resource Isolation ", Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS ' 02)
- [4] P. Gupta and N. McKeown, " Packet Classification on Multiple Fields " , Proceedings of ACM SIGCOMM ' 99, Cambridge,MA, September 1999.



Cheng Chuanhui (1979-?) is a doctor candidate of Wuhan University of Technology, major in high performance network.

A Hierarchical System Management Approach Based on SNMP for Network Device

Yan Bin , Yang Zijie

School of Electronic Information, Wuhan University

Wuhan ,Hubei 430079, China

Email: lzheng_2000@sohu.com Tel: 13971562055 (027) 62485381

ABSTRACT

System Management(SM) architecture of network device can use a hierarchical management approach. The system has standardized SNMP interface to the external management subsystem. Configuration functionality, status query functionality, statistics collection functionality and event reporting functionality for its all modules can be provided by this hierarchical management approach. More dynamic extensibility can be obtained and the network device can be easily managed by other external management subsystem.

Keywords: SNMP; hierarchical management; system Management; external management subsystem

1. INTRODUCTION

System Management(SM) architecture of network device may use a hierarchical management approach. After this approach is used, the system has standardized SNMP interface to the external management subsystem (EMS). The approach has dynamic extensibility and can be applied in complex network device. There is a central point of control and management of the complete system that can be easily realized system level management [1].

2. TOTAL CONSTRUCTION

System Management(SM) architecture of network device can be described as Figure 1.

An SNMP agent can be plugged in between the EMS and the network device. There exists a Central SM agent for the whole device. And there exists a Local SM agent on every board instance.

The SNMP agent communicates with the Central SM agent within the device using well-defined backend APIs. The Central SM agent is then responsible for exchanging messages with appropriate modules or the Local SM agents of the device. The Central SM agent will provide for the system level module management aspects of the device, while the Local SM agent will provide for the instance level module management aspects of the device.

When the Central SM agent receives a message from the SNMP agent, the Central SM agent will serve the message if the message is pertaining to system level module management. If the message is pertaining to the instance level module management, the Central SM agent will send the message to the appropriate instance level Local SM agent, which will in turn process the message appropriately.

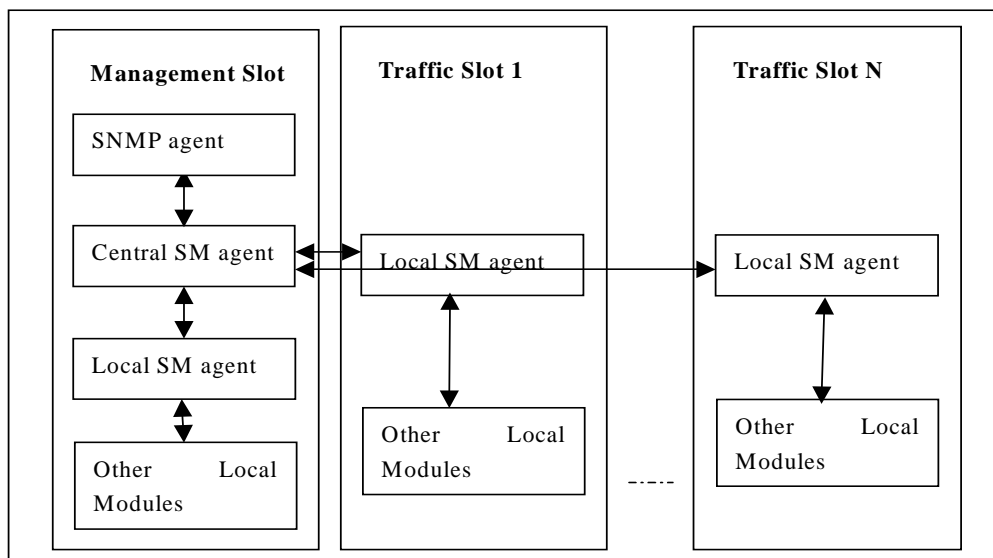


Figure 1 System Management(SM) architecture of network device

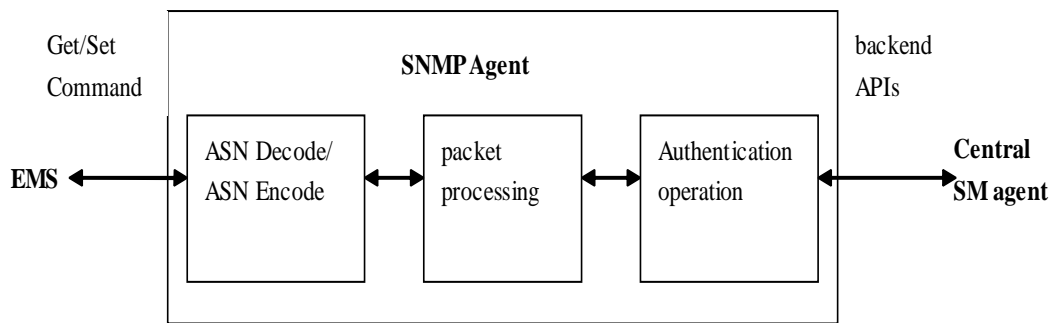


Figure 2 architecture of SNMP agent

3. ARCHITECTURE OF SNMP AGENT

The architecture of SNMP agent can be described as Figure2

An SNMP agent will be running which receives a SNMP request from the external management subsystem (EMS). After receiving a SNMP request, the agent groups the OIDs (Object Identifier) in the Varbinds in accordance with their access group since get or set method routines are provided at access group level.

In the access functions user code for getting/setting the value of the parameter in the system will be written. Corresponding to each SNMP set/get commands backend APIs will be invoked by the SNMP agent towards the Central SM agent. The Central SM agent shall send the appropriate command to the respective managers, which shall process the command. The SNMP agent after receiving the response from the central SM agent will send the SNMP response back to requesting EMS [2].

All the traps generated in the device will be forwarded to the SNMP agent from the Central SM agent using the backend APIs defined for the trap. The SNMP agent will forward the trap to the EMS. The basic steps performed by the SNMP agent to process SNMP messages receives from the EMS are given below: upon receipt of a request from the EMS, the packet is ASN decoded and the system dependent packet processing is performed. Upon the completion of system dependent packet processing, the entire authentication related operations are performed and then requested operation will be carried out. SNMP agent's ASN.1 encoded response will be sent as a reply when the corresponding module responds.

4. SNMP INFORMATION FLOW

Standardized SNMP interface can provide some system management functionality such as: configuration functionality, status query functionality, statistics collection functionality and event reporting functionality [3].

4.1 Configuration information flow

The configuration functionality at both the Central SM agent and the Local SM agents will be responsible for configuration of the network device. The startup configuration of the network device shall be performed as defined below:

The configuration functionality of the Central SM agent will read the specified configuration file at the startup of the system and configure the system and pass the configuration data of individual instance to respective Local SM agents. The configuration functionality of the Central SM agent will store the global data for the complete configuration of the network device.

The configuration functionality of the Local SM agent will send the configuration APIs to the appropriate module. It shall locally store the slot level configuration data in its local memory. This does not include the configuration data of individual module, which will be stored within the module's own memory. For the configuration parameters whose values are not supplied from the configuration file, default values will be used.

4.2 Status Query information flow

The status query functionality, in both the Central SM agent and the Local SM agent, shall be responsible to respond to the query of network device and its modules. The status query functionality of the Central SM agent will respond to any system level query requested by the user. For a specific instance-level query the Central SM agent will pass the message to the appropriate Local SM agent and receive the response.

The status query functionality of the Central SM agent as well as the Local SM agent shall also respond to any inventory related queries from the user. The inventory related queries should contain the information specific to version number, instance number etc.

4.3 Statistics Collection information flow

The statistics collection functionality of the Central SM agent and the Local SM agent shall be responsible to collect and send the statistics to the EMS for the network device and its modules on request. The statistics managers can be optionally configured from the user/configuration file to collect statistics from the system and its modules.

The statistics collection functionality of the Central SM agent will respond to any system level statistics requested by the user, either by collecting the statistics at the time of data request or returning the data collected over a period of time.

If the statistics requested is instance-level, the request will be passed to Local SM agent and the data will be returned to the statistic collection functionality of the Central SM agent

from the statistic collection functionality of the Local SM agent. The statistics collection functionality of the Central SM agent will also be responsible for collating the statistics data and sending a collective response to the user, in cases of module level statistics data request where module may be present on more than one slot.

4.4 Event Reporting Functionality

The event reporting functionality will reside at the system controller instance in the Central SM agent. The main responsibility of this functionality is to relay the traps received from the modules and the fault manager to the element manager via the SNMP agent. The Central SM agent can be configured to filter the traps before reporting to the element manager. It shall allocate a unique trap id to each trap sent to the element manager.

Any event occurring in a module will be directly notified to the event reporting functionality of the Central SM agent. All the faults (alarms) raised by a module will be reported to the corresponding fault manager. The fault manager, if required, will collate a set of faults raised and will report a single fault to the Central SM agent's event reporting functionality.

The event reporting functionality will be configurable to store a defined number of traps reported to the element manager as traps. This will be used for reconciliation of alarms from the element manager. The event reporting functionality will provide and interface to fetch a particular trap notified to the element manager on request from the EMS.

In addition to notifying the traps to the element manager, the event reporting functionality will also be responsible for logging. All the modules shall send the log request to the event reporting functionality in a well-defined format. If the modules are resident on instances other than the system controller instance, the event will be reported via the Local SM agent to the event reporting functionality of the Central SM agent. The event reporting functionality will be configured with a base log-file name, which will be created during the runtime timestamp. The limit of the log file size will be a configurable parameter. When a log file is full, it will be stored onto a persistent storage and a new log file will be created with the new creation data and timestamp in the log file name.

5. MIB DEFINITION APPROACH

The SNMP can support configuration of a single configuration parameter or a group of configuration parameters defined within the same MIB. However the protocol APIs provided by the modules may expect a group of configuration parameters, some of which may have multiple instances within the same protocol module API. It cannot be assumed that all the configuration parameters are provisioned from a single SET request [4].

The SET requests received over the SNMP shall be dispatched to these modules using the protocol APIs. For a single protocol module API, the configuration parameters may be received across different SET requests.

To address this issue, MIB groups will be defined as the module configuration APIs. The element manager will ensure that in a single SET request all the parameters of the MIB group shall be sent.

For the protocol module APIs in which the multiple instances are present each instance can come in a separate SET request. SET requests for different protocol module API for the intended module/module shall be collated using an Enable/Disable flag. This flag shall initially be set to disable. When the flag is set to disable all the SET requests corresponding to a single protocol module API shall be dispatched to the target module/module. There shall exist an adaptation layer, which shall store these values. When the enable flag is set, all the configuration parameters corresponding to the protocol module API shall be validated. If the validation succeeds, the protocol module API shall be constructed and then dispatched to the module/module core.

The enable/disable flag will also be present in the MIB groups for which the parameters contained do not have a default value defined or which can be initialized only when the configuration is available. This ensures that the SET requests received from any element manager will only be committed to the modules if all the information has been received corresponding to the MIB group.

If a SNMP SET request is received for a MIB group whose Enable/Disable flag is set to enable, the adaptation layer shall construct the appropriate protocol module API and send to the protocol module. If the protocol module API cannot be dispatched again to the protocol module, the adaptation layer will return error in the SET response.

All MIB groups will also contain a create/delete flag, which shall ensure that only valid entries are accepted by the network device. Whenever a SET request is sent for any instance of the MIB group to the network device, it is necessary to first send the create/delete flag for the corresponding instance id for the MIB group. If a SET request is received for a MIB group instance from the element manager for which a create/delete flag is not set to "created", the network device shall return a failure.

An element instance identifier will identify each system controller instance, signaling instance and the traffic instance. These instances will host protocol modules, each one of which is identified by a module instance identifier. In the MIB there will be a MIB group containing the mapping of the element instance identifier to protocol module instance identifier. Based on this mapping the protocol module identified by the protocol module instance identifier will be resident on the element instance to which its mapping is created. Both these instances will be unique across the network device [5].

This mapping table will be useful to handle redundancy and the operator initiated switchover procedures when a protocol module instance may be moved from one element instance to the other. During these procedures the mapping of the protocol instance identifier to the element instance identifier will be updated to the new element instance hosting the protocol module. However the protocol module will retain the same configuration as what existed on the old element instance.

6. CONCLUSION

Through the interaction of the SNMP agent, Central SM agent and Local SM agent, the complex network device can

provide a hierarchical management interface to manage its all slots and modules. Configuration functionality, status query functionality, statistics collection functionality and event reporting functionality for its all modules can be provided by this hierarchical management approach. When a new slot is added, module management of other slots can't be changed. So more dynamic extensibility can be obtained. Furthermore, the network device can be easily managed by other users and external management subsystem with standardized SNMP interface.

7. REFERENCES

- [1] WU Li-fa. Researches on The Distribution Policies of SNMP-based Network Management System[J]. *Telecommunication Technology*, 2001
- [2] IETF RFC1157-1990. Simple Network Management Protocol [S].
- [3] CHEN Xian-dao, AN Chang-qing. SNMP and its Application Development [M].
- [4] ZHAO Xiao-rong, NV Bing, SONG Kai. Analysis and Access Implementation of MIB Based on SNMP Network Management Protocol [J]. *Computer Exploitation and Implementation*, 2001 Beijing: Tsinghua University Press, 1998
- [5] IETF RFC1213-1991. Management Information base for network management of TCP/IP-based Internets: MIB-II[S].



Yan Bin is a Doctor Student of School of Electronic Information in Wuhan University. She is graduated from Wuhan University in 1997 and gets a Bachelor Degree with specialty of electronics and information she is a Master Student in 2001 and Doctor Student in 2003 with specialty of system.

And she is a Master Student in 2001 and Doctor Student in 2003 with specialty of radio physics. Her research interests are radio based on software, mobile communication, and computer network.

VLAN Aggregation Technology Research and Implementation

Cheng Chuanqing

Computer Science Department, Wuhan university of Science & Engineering

Wuhan, Hubei, 430074, China

Email: ccq@wuse.edu.cn Tel: 86-27-62821633

ABSTRACT

This paper gives the details of Vlan Aggregation technology and explains not only the concept of it, but also explains it works in a embedded operating system-vxWorks. This paper also discusses Layer3 switch and network vlan, puts forward that the layer-2 vlan role and layer-3 vlan role. We introduce the notion of sub-VLANs and super-VLANs, a much more optimal approach to IP addressing can be realized. This paper will expatiate the VLAN aggregation model and its implementation in Ethernet access network. It is obvious that the customers in different sub-VLANs cannot communication to each other because the ARP packet is isolated. Proxy ARP can enable the communication among them. This paper also expatiates the proxy ARP model and its implementation in Ethernet access network.

Keywords: Vlan supervlan subvlan network interface proxy arp

1. INTRODUCTION

While the Ethernet is applied in public network environment, people demand much higher security. The development of vlan restrict the broadcast domain, to some extent, which improves the security of Ethernet. With the development of switch Ethernet, layer-3 switch is applied more and more, which has almost replaced router at the edge of the network. In the Layer-3 switch architecture, hosts in the same vlan communication each other via layer-2 switch mode, forwarding packet via destination Mac address, while hosts in the different vlan communication each other via layer-3 switch mode, forwarding packet via destination ip address. In this way, vlan means a broadcast domain in layer2 and means a network interface, a subnet, and a gateway of the hosts in the vlan in layer3. Obviously we hope that the broadcast domain is smallest and the occupied network address is fewest, which is a big conflict in network layout. The main aim of Vlan Aggregation is to improve Ipv4 network address utilize rate. Especially with the campus access network or ISP environment, VLAN Aggregation can make the hosts of the same IP subnet stand in different broadcast domain and share the same gateway.

In the VLAN Aggregation architecture [1], a supervlan can be denoted a ipv4 address while a subvlan can not. But all the ports that link with the hosts are attached with the subvlan. All the hosts of this subvlan use the same network segment address and mask with the supervlan and make the supervlan's ipv4 address as the gateway.

Hence, the broadcast of subvlan can be most small, extremely include only one host, which is very useful in campus access network environment. Customers of the campus network hope protect private data, so the port isolate is necessity, old vlan architecture will greatly waste ipv4

resource. Vlan Aggregation is applied in this environment. While we break a subnet into different broadcast domains (subvlans), the hosts within different subvlans communication each other via proxyarp. [2] The supervlan makes the hosts in different subvlan a "virtual broadcast domain" by proxyarp. Host route module is responsible for the following data forwarding. Figure1 shows Vlan Aggregation architecture.

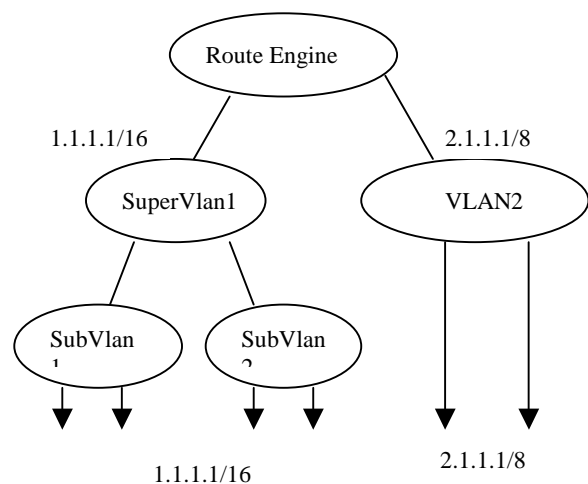


Figure 1 Vlan Aggregation architecture

This paper is structured as follows. Vlan, layer-3switch and net interface concept will be introduced in section 2. The related technology and implementation in vxWorks of Vlan Aggregation is described in section 3. How the arp proxy makes the different subvlan communication each other is described in Section 4. Finally, a summary of the paper content is provided in Section 5.

2. VLAN, LAYER -3 SWITCH, NET INTERFACE [2][3][8][9]

The network VLAN is the key of Layer-3 switch. The key is that a vlan must be a subnet. Perhaps someone will think of multi-segment in a vlan. It is possible but not recommend. According to IP forwarding rule, the hosts which belong to different subnet can not communication in layer2, but in layer-3. we can think it link this: even if a broadcast domain have more than one subnet, hosts in different subnet also communication in layer-3. But the layer-2 broadcast packets can forward in different subnets, which will occupy the limited bandwidth. So typical configuration is a vlan, a subnet, just like Figure 2.

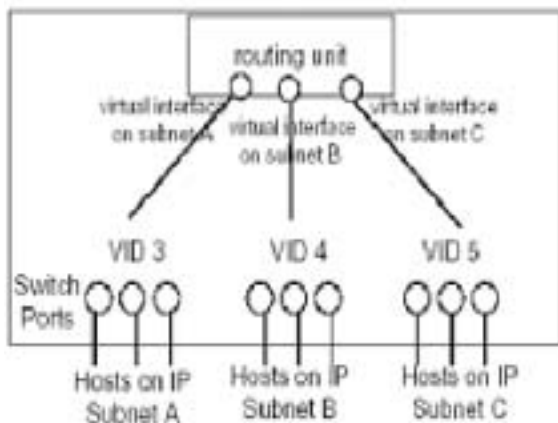


Figure 2 a Vlan,a subnet

So we can make each vlan corresponding to a virtual interface. Each interface has own ip address and Mac address and ip address.

Each network vlan has the following feature [6]:

- Each vlan has a related Mac address, and can be set an ipv4 network address and a mask.
- Each vlan that is set an ipv4 network address/mask can be a network interface, which is bound with the OS protocol stack. The subnets of this vlan can communication to the end-station inside via the network interface.
- The ipv4 address /mask of the vlan is set by the host as the default gateway.
- The vlan can isolate layer2 broadcast domain, the broadcast packets and the unknown packets will flood within the vlan but not to other.
- According to the practical need, the end stations in different vlans can be set to communication each other or not
- Each vlan, which is set an ipv4 network address/mask, is bound with OS protocol stack as an independent device in the low layer driver, and responsible for process and reply the communication with the ip.

3. VLAN AGGREGATION TECHNOLOGY

With Vlan Aggregation technology, we introduce the notion of subvlans and supervlans, a much more optimal approach to IP addressing can be realized. Essentially, what occurs is that each sub-VLAN (customer) remains within a separate broadcast domain. One or more subvlans belong to a super-VLAN, and utilize the default gateway IP address of the super-VLAN. Hosts within the subvlans are numbered out of IP subnets associated with the super-VLAN, and their IP subnet masking information reflects that of the super-VLAN subnet. So the separate broadcast domain is corresponding to subvlans and the IP sub network is corresponding to supervlans.

3.1 Subvlan [7]

Sub-VLAN is a relative notion, contrast to super-VLAN. To understand subvlans without layer3 conception, the sub-VLAN is Virtual LAN. A sub-VLAN means a layer2 broadcast domain, the broadcast information only transmitted in the sub-VLAN. In many switches, VLAN's

implement is by ASIC; a subvlan may occupy a vlan entry of the chip, as a vlan .It is a very important conception that in layer 2,or before a "sub-VLAN" is added into another vlan (super-VLAN), sub-VLAN is vlan.

However, view at layer 3, there are some differences between sub-VLAN and ordinary VLAN. The most important difference is that a sub-VLAN cannot directly correspond to an IP subnet, but an ordinary vlan can .We cannot configure an IP address to subvlan as the host's default gateway. The following is that sub-VLAN usually consists of fewer hosts than ordinary VLAN, because several subvlans can aggregate into a supervlan.

3.2 Supervlan [7]

On the contrary, the super-VLAN is a logical concept. It does NOT mean a real VLAN, but a virtual device interface. The main function of the super-VLAN is supply the default IP gateway for the hosts of per sub - VLAN who belong to the super-VLAN.

Not as sub-VLAN, supervlan don't occupy a vlan entry of chip. We cannot understand super-VLAN as the ordinary VLAN. When understanding supervlan, it is wrong that "port based super-VLAN" or "802.1Q based super-VLAN". In fact, a super-VLAN can only consist of subvlans, but not switch ports. The super-VLAN merely borrows the name of the "VLAN", but is NOT a real VLAN.

4. VLAN AGGREGATION IMPLEMENTATION

Vlan Aggregation implementation goes by the flowing key steps:

4.1 Create vlan

Either supervlan or subvlan is an ordinary vlan at the beginning of creation. They both have a way from vlan to END driver []. The data structure of vlan is following:

```
struct vlan
{int vid,
...
int flag;
int parent;
void *netdevice;
}
```

4.2 Being supervlan or subvlan

According to network management module, we can set the relation between two vlan, make one as supervlan and the other subvlan. Then we can change the corresponding data structure, not only to identify the feature of vlan but also to get the relation between supervlan and subvlan. An example data structure is like this:

```
Supervlan
{ vid = 1
flag = SUPRE;
parent = 0;
void *netdevice = addr1;
}
Subvlan
{ vid = 2;
flag = sub;
parent = 1;
```

```
void *netdevice = addr2;
}
```

4.3 Low layer data receive algorithm

After driver receives data, it will send it to the vlan-related END driver. Because the subvlan only undertake the task to restrict broadcast domain, the function of the network vlan must be finished by supervlan. A algorithm is applied to reach it.

```
if (netdevice->vlan is subvlan)
{
    supervlan = getvlanbyvid(vlan->parent);
    dev = supervlan->netdevice;
}
}
```

4.4 Low layer data send algorithm

The flow of IP layer sends packet is more complex. IP layer utilizes END driver to send packet. The send algorithm is like this:

```
If (unicast ip address)
{
    get the related supervlan via END
    get the subvlan accroding via mac in fdb;
    if (success)
        give parameters(data,subVID) to driver,driver doesn't see
        sueprvlan;
    else
        drop the packet;
}
else
{
    send packet to every subvlan of the supervlan;
}
}
```

Figure3 shows the relation of TCP/IP stack,vlan,and net device.

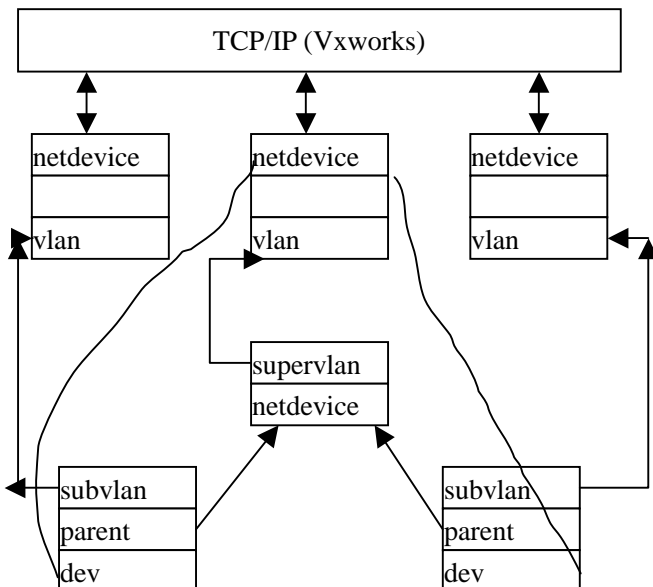


Figure 3 TCP/IP, net device, vlan

5. PROXY ARP [4]

Proxy arp means a procedure like this: end station in a

broadcast domain send arp request to the end station in the other broadcast domain, because the arp broadcast packet can't penetrate different broadcast domain, the gateway will reply the arp request, which means proxy arp. In the following communication, sender will send packet to gateway, it is the gateway that undertake the layer-3 forwarding task.

In the Vlan Aggregation architecture, an IP subnet (supervlan) can consist of different broadcast domains (subvlans). The Arp request packet from one subvlan to another within the same supervlan will broadcast only in the own subvlan, the destination end station will not receive the Arp request. So the route node of different broadcast domain, supervlan, and supplies the proxy Arp, as a gateway, is responsible for Arp among different subvlan.

The proxy Arp algorithm in Vlan Aggregation is following:

- 1 if gateway finds that the destination is reachable, but the source station and the destination station are in the same subvlan, gateway doesn't process the Arp request, let it processed by end station itself.
- 2 if gateway finds that the destination is reachable, and source station and the destination are in the different subvlans, gateway replies the Arp request by its own hardware address.
- 3 if gateway finds that the destination is unreachable, it tries to find the destination, then process it as 1 or as 2.

6. CONCLUSIONS [9]

With the campus network environment, customers wish to be isolating each other to protect their private information from sniffing. Mini vlan, which include only one port, can meet this. But the following trouble of too many ip network disturbs the network lay out. This paper proposed Vlan Aggregation to solve this problem ,gives the implementation of Vlan Aggregation in layer-3 switch fabric and describes concrete receive and send algorithm. Proxy Arp algorithm is also described. The architecture has been applied in DSL Multiplex device.

7. REFERENCES

- [1] D. McPherson, Amber Networks, Inc. B. Dykes Onesecure, Inc. VLAN aggregation for efficient IP Address Allocation.RFC3069, 2001.
- [2] COMERDE.Internetworking With TCP IP Vol1.I [M]. Beijing: Prentice-Hall, 2000
- [3] XiRen Xie, Computer network,[M],Beijing, electron industry publication,1999
- [4] J. Postel, Multi-LAN Address Resolution.RFC925, 1984
- [5] LISTANTIM. Architectural and Technological Issues for Future Optical Internet Network [J]. IEEE Comm. Magazine, 2000,38(9): 82-92
- [6] He Xiaoming, "Analysis of layer-3 switch technology", China Data Communications, 2003
- [7] Li Wang," Security management of Broadband Access Network based on Vlan Aggregation", Asia-Pacific Optical and Wireless Communications 2003
- [8] G.R.McClain," Hand Book of Networking and Connectivity" AP Professional, Boston, MA, 1994.

- [9] J.Gong and P.Srinagesh,"An Economic Analysis of Network Architecture," IEEE Network, vol.10, no.2, March/April 1996,pp.18-21



Cheng Quanching is an engineer of Wuhan University of Science & Engineering. He engages in the field of network layout, design, and implementations His research interest also includes network protocol study. He graduated from Wuhan University in 1996 with specialty of electronic & information system. He has published

several papers and taken part in compiling more than one book.

Study and Application on Time Synchronization Technique in Computer Network

He Peng

Information Technology Center, Three Gorges University
Yichang, Hubei 443002, China

Email: hpeng@mail.ctgu.edu.cn

Xu Yishan, Tao Ke, Dai Hui

College of Electrical & Information, Three Gorges University
Yichang, Hubei 443003, China

Email: keketao007@163.com

ABSTRACT

Time is a basic physical variable and one of the most important basic data in computer network. Time synchronization technique in computer network has become one of the most important problems. On the basis of the analysis to existing achievements, the paper has introduced various techniques about time synchronization, proposed solutions in different application backgrounds, and gave application analysis and conclusion.

Keywords: Computer Network, Time Synchronization, Network Time Protocol, Time Server.

1. INTRODUCTION

At present, GPS, which provides precision better than 1 microsecond, is mostly applied in cross-time zone, global and high precision time synchronization. It is widely used in scientific studies and engineering, such as spaceship launch, satellite measurement and control, weather forecast, earthquake prediction, etc. Taking national security into account, china still uses its own BPL and BPM time service system. With the improvement of technology and the enhanced requirement of time precision, various new transmission techniques of time are gradually being deeply studied and widely applied. According to the affirmation of national time service authority, time users asking for middle and low precision is increasing rapidly and has become the majority. The number of users asking for precision in the order of 1 second is 3 times more than that of 1 millisecond, while the number of users asking for precision in the order of 1 millisecond~1 second is 9 times more than that of 10 microseconds. New methods, such as Computer network time service, telephone time service, TV time service, etc, could provide service with middle and low precision. Among these methods, the study of computer network time service is the hot spot.

NTP (Network Time Protocol) earlier proposed the idea of providing time service based on computer network [1]. However, NTP is not that usable, and it doesn't refer to concrete algorithms of Time Server and time user. Based on this, in references [2][3], the author respectively introduces time synchronization algorithms in distributed system and in WAN, and presents applications of some algorithms in SCADA (Supervisory Control and Data Acquisition) of power plant [4] and the design and implementation of synchronization devices [5]. This paper mainly discusses various techniques and approaches of computer network time

transmission, proposes implementation solutions in 2 different application backgrounds, and analyses a typical application.

2. TIME SYNCHRONIZATION TECHNIQUE

Active Synchronization and Passive Synchronization

Every time application system should have the capacity of maintaining time and administrating its performance. The only approach for user to achieve time information is to visit time maintaining architecture- the internal reference time of the system. Apparently, during the visit, user synchronizes his clock with the internal reference time. Since there is no connection between the internal reference time and external time, the synchronization is limited only inside the system. This kind of synchronization is called passive synchronization or relative synchronization. However, the time maintained by internal reference time cannot reach absolute rapport with the physical time maintained by UTC, with the result that passive synchronization is available inside the system but cannot be used outside. In order to have all the users' clocks inside the system synchronized with UTC, it's necessary to manage to synchronize the internal reference time with UTC initially. This kind of synchronization is called active synchronization or absolute synchronization.

Because UTC (Coordinated Universal Time) is in rapport with TAI (International Atomic Time), which, published by International Time Administration, is a weighted average of over 100 values from atomic time scales distributed in 25 countries and all official standard time systems take UTC as their time reference, UTC is called absolute time, and in research it's also called physical time.

High accuracy time signal from GPS is maintained by 2 rubidium clocks and 2 cesium clocks carried by satellite and it keeps synchronous with UTC through ground control station. Practically, time signal from GPS has been regarded as PTS (Primary Time Standard) of the most time applications in the world. So, to make absolute sense, studying active synchronization appears to be necessary when we study network time synchronization. In this paper, Time Server refers to the architecture in computer network, which can maintain time and keep its stability.

Time Transmission Approach

To transmit PTS signal from GPS to user through Time Server in computer network, there are mainly two steps: 1, directly transmit time signal from GPS to servers; 2, Transmit time signal from server to user through network

protocols.

Transmission from GPS to Time Server: Direct time transmitting technique includes 6 approaches subject to 3 types: The first is coding type including COM RS232C time coding and IRIG time coding. The common characteristic of them is that time information, such as year, month, day, minute, second, etc, is transformed and added into level bit and byte for transmission in the form of binary, BCD or ASCII. Asynchronous transmission mode and standard connection interface make it relatively convenient. The second is pulse type mainly including 3 modes: 1pps (pulse per second), 1ppm (pulse per minute) and 1pph (pulse per hour). All of them are periodic and have stringent timing requirements on increase width and pulse width. Furthermore, increase width must be stringently synchronized with UTC in the precision superior to 1 microsecond. The last is frequency reference signal, which generally appears to be a kind of concomitant modulate signal.

Figure 1 illustrates time coding and signal format of 2 connection modes of direct time transmitting technique (more details are in table 1), which is characterized by its optimal precision and is mostly implemented in engineering field [6].

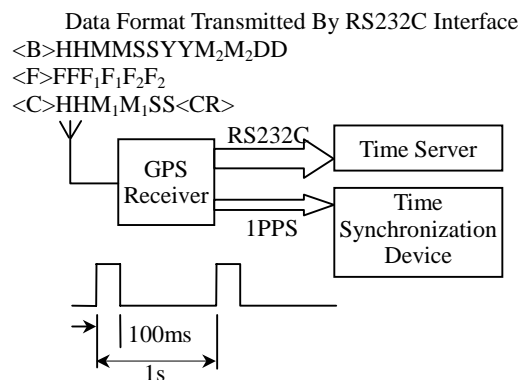


Fig.1 Direct Time Transition and Signal Code Format

Table 1 GPS Receiver Port and Signal Type

Port	Interface Type	Signal Type
Input Port	Time Code Input (RS232C) 1pps(1ppm, 1ppm selective)	Coding Type (BCD, ASCII) Pulse Type
Output Port	Time Code Output (Selective) 1pps(1ppm, 1ppm selective) Frequency Flag Output	Coding Type (BCD, ASCII) Pulse Type Modulation
Network Port	AUI Ethernet interface (Selective) 100Base-T Ethernet Interface RS232C	Coding Type

Transmission from Time Server to User: The transmission technique based on network protocols is used when time information travels from server to user units through network. Since network structure and its protocol topology are prior assigned, we realize the transmission between servers and users mainly through NTP.

NTP, defined by a set of RFC (Request for Comments) documents of network time synchronization, is a group of standard network protocols, which focus on synchronization calculations [7]. Among the protocols, TP and DTP are comparatively simple which can meet the requirements of most users by providing 1 second precision. As to the users who request higher accuracy, NTP is able to achieve a precision of 1~50 millisecond. Generally, DTP and NTP can satisfy most time users asking for different precisions. SNTP, the simplified edition of NTP, provides a slightly lower precision than NTP does.

Time Synchronization Algorithm

Time synchronization algorithm describes the process of realizing synchronization in computer network- a time application system, and it mainly reflects the synchronous relationship between Time Server and server, server and time user. Essentially, this algorithm commits to figure out transmission delay of time signal and rectify the delay.

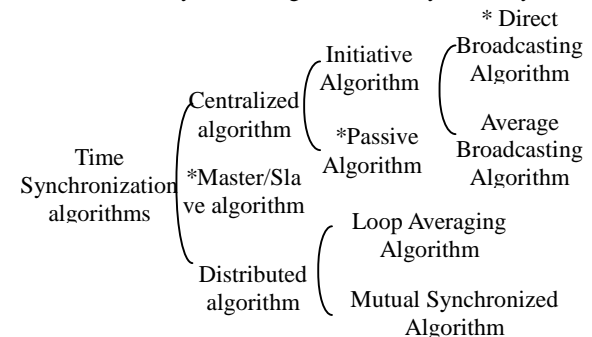


Fig. 2 Classification of Time synchronization Algorithm

Figure 2 presents a simple classification of synchronization algorithm. Theoretically, all algorithms are available under passive condition. But, if taking algorithms' different characteristics into account, the algorithms with "*" in figure 2 would achieve better performance under the active condition. Accordingly, passive application of algorithms with "*", which has excellent short-term stability, is used as a backup for active application.

Centralized Algorithm: This kind of algorithm is characterized by assigning a host node in computer network (especially in LAN) as Time Server. It can be classified into initiative algorithm and passive algorithm, according to whether server periodically broadcasts time actively in network or server offers time only to those who make a request.

In direct broadcasting mode, Time Server takes the minimum clock periods of all the user nodes in the network as its own period to broadcast time information. The information travels through network and reaches every user node with different delays. Users pick up the information, deduct transmission delays and calculate their time. Affected by the uncertainty of transmission delay, in this mode synchronization can achieve a precision of 10~100 milliseconds.

Instead of broadcasting the server time, in average broadcasting mode, the time broadcasted by server is an average of all the users' time acquired in the manner of server inquires and user answers. Meanwhile, the server adjusts its own time to the average value too. Average broadcasting mode has remarkable superiority under passive condition.

When periodically broadcasting, initiative algorithm does not distinguish relative brushing period (to the Time Server) between client nodes. Passive algorithm is different. Client node decides whether it should ask server for time information to reach synchronization according to the required precision and its relative brushing period. Since the server responds only when being asked, the algorithm is very applicable to be realized in Client/Server mode.

Figure 3 shows how to estimate delay in passive algorithm in sketch. Assume t_1, t_2, t_3, t_4 are respectively the time when user sends request, server receives the request, server responds and user receives the response. Then, the round trip

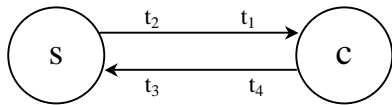


Fig. 3 The Estimated Value of Transmission Delay of Passive Algorithm

delay is: $\tau = (t_4 - t_1) - (t_3 - t_2)$. And assume the network is symmetrical, in other words, the time delays spent on the way to and fro are the same. The single trip delay should be $\tau/2$. We can conclude that the reference time received by user is: $t_c = [(t_1 + t_4) - (t_2 + t_3)]/2$. Since the algorithm is in Client/Server calculation mode and can be with NTP protocols if necessary, it can realize the precision of 1~10 milliseconds.

Master/Slave Algorithm: Generally, Master/Slave algorithm is applicable for mesh or tree architecture, and tree architecture is the majority in practice. In tree architecture, master Time Server is located at root node, which can either connect to the external time source (for example GPS) with direct time transmitting technique to realize a stable and high accuracy synchronization, or take its own time as the reference time of the whole time application system. In the course of synchronization, the time signal from master server is directly transmitted to sublevel server. Repeatedly, the sublevel server transmits the time to its sublevel. As presented in figure 4, the synchronization of the whole network can be realized in this level-to-level way.

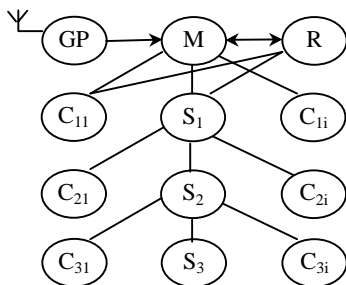


Fig. 4 Principle of Master/Slave Algorithm

For other applications with higher accuracy requirement, such as telecom synchronization network, to guarantee the reliability, backup server is set up at root node. When master server works well, the backup is under the control of the master and is synchronized with the master. When the master

suffers a failure, the backup takes the control until the master resumes. Assume $T_m(t)$ is the reference time from server,

ΔT_p is the deviation between 2 neighboring level servers, and tp is the transmission delay between the 2 servers. Then we can easily figure out that the time value of the server at N levels is:

$$T_N(t) = T_m(t) + \sum_{p=1}^{N-1} (\Delta T_p + tp)$$

Distributed Algorithm: Centralized algorithm and master/slave algorithm both rely heavily on Time Server, and this is their distinct defect. We can use distributed algorithm to synchronization more reliable. Its characteristic is: It achieves the reference time of network through designing and executing algorithm instead of does so through a single assigned Time Server. It can be divided into loop averaging algorithm and mutual synchronized algorithm.

The main idea of loop averaging algorithm is: In a network with M host nodes, we repeatedly assign N ($N \leq M$) nodes as Time Servers, and execute the loop averaging algorithm on the N servers to achieve a reference time, which will be broadcasted later to synchronize all the nodes in the network. Such repeated average calculation and broadcast keep the time of the whole network in a "medial" state. The time is called "network time" or "network frequency".

To make it easier, among the assigned N nodes, each time we select 2 nodes and do synchronization. Then, under a certain rule we repeat the synchronization in pairs and finally achieve the final synchronization among all the N nodes. Obviously, although the execution of the algorithm differs from loop averaging algorithm, the final result of the algorithm is still a kind of "network time". And this algorithm is called mutual synchronized algorithm.

3. TIME SYNCHRONIZATION SOLUTIONS

Independent Architecture

This solution is put forward mainly for the single close computer network. It consists of 4 parts: GPS spatial part, GPS receiver, Time Server and computer network. As shown in figure 5, the main body of GPS spatial part is GPS satellite, which is in charge of providing visual time signal synchronized with UTC. GPS receives, processes and separates the signal, and finally provides 3 types of signals for output (as mentioned above). Time Server can be either a special host computer, or an assigned one in network. Interface types in Time Server and GPS receiver are listed in table 1.

Computer network is a kind of application system, which focuses on the synchronization algorithm based on protocols such as NTP.

Independent architecture solution for synchronization has many merits, such as simple architecture, low consumption of network resource, etc. Meanwhile, it also has defects such as no centralized supervision and network administration function because of the independence among its inside architectures.

Simple Network Administration Architecture

As shown in figure 6, to improve the cooperation and to

enhance the capacity of centralized supervision over network, which is just the defect of independent architecture solution, simple network administration protocol can establish connection between independent architectures and realize the

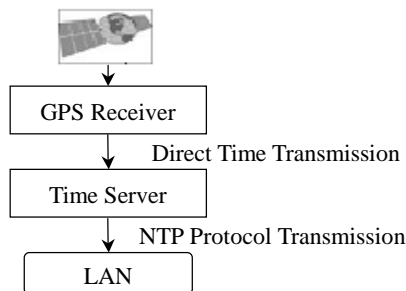


Fig. 5 Independent Architecture

centralized supervision and administration. Simple network administration architecture has such merits: It has a time source which is more uniform and can provide higher accuracy; the centralized supervision and network administration function can provide better coupling and better resolution of time for time application system. But, it has the defect of large investment on equipments.

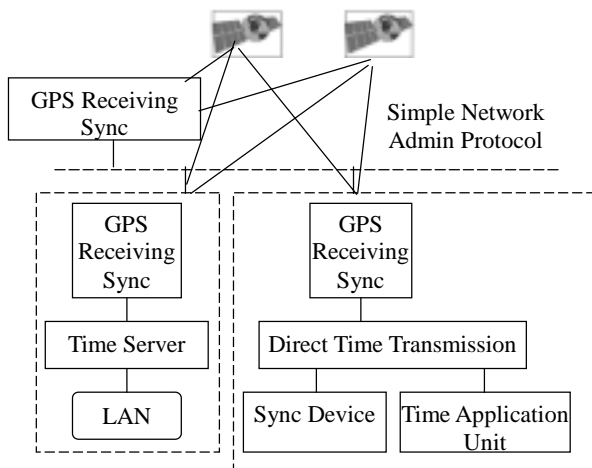


Fig. 6 Simple Network Admin Architecture

4. AN APPLICATION AND THE ANALYSIS

Figure 7 presents us a time service model based on C/S (client/server) pattern, which is practically used in SCADA system of Gezhouba dam power plant. In the system, we adopt two cooperative and mutual-backup work modes: one is active absolute synchronization mode and the other is passive relative synchronization mode. In the former pattern, the GPS receiver transmits a time code to the Time Server periodically (hourly here) with a deviation of less than $1\ \mu\text{s}$ to UTC. Then, the server receives the code and synchronizes its own clock. Finally the server offers calibrated time information as a reference to the client according to its request. In the latter pattern, there is in no need of signals from GPS receiver. When a client makes a request, the server transmits un-calibrated time information maintained by itself as a reference to the client. Obviously, the active absolute synchronization is based on UTC, while the passive relative synchronization takes server's time as its criterion. In normal situation, the first pattern is used in the system to achieve the synchronization, while it switches to work in the latter pattern

only when GPS signal is not stable or even lost. Actually, this backup manner is a necessity to gain a stable performance of the application system.

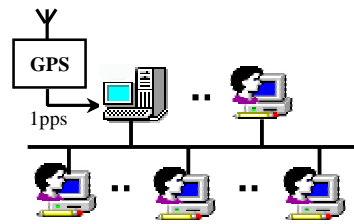


Fig. 7 Topology of Time Service Application Based on C/S Mode

Figure 8 illustrates a comparison between the two patterns above by their statistical average errors measured every day between 8:00 AM and 22:00 PM, 5/8/2002 to 5/24/2002. Obviously, since the time is synchronized against UTC in 1 hour's interval, in the active pattern, the change of errors is comparatively slight and value of them is below 10ms. While in the passive pattern, because no GPS time code is received after being synchronized with UTC at 8:00 AM, the value of errors' change can maintain below 10ms during the first 5 hours. After that, since the deviation between the Time Server and UTC is getting large, the enhance trend of errors is becoming large too. From the figure we can draw three conclusions: 1 It is better to use active pattern when high synchronization precision is required. 2 Passive pattern can be used as a short-term backup for active pattern. 3 Two patterns can cooperate and mutual-backup for each other.

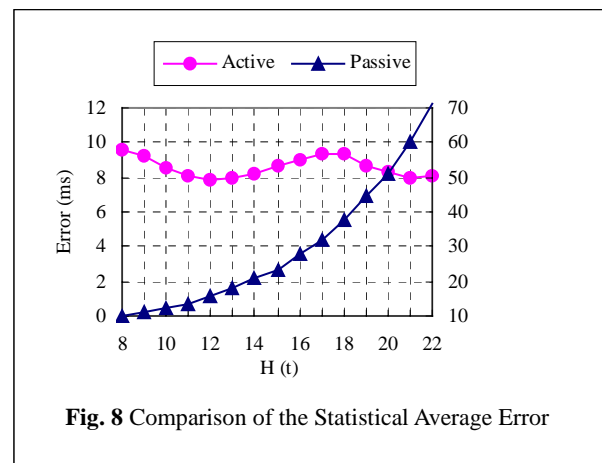


Fig. 8 Comparison of the Statistical Average Error

5. CONCLUSIONS

This paper specifically analyses and summarizes various technologies, solutions and applications adopted in the area of time synchronization in computer network. It's reasonable to conclude that choosing a proper synchronization algorithm, which actually focuses on how to manage transmission delays in network, is the key to realize high accuracy synchronization in computer network. With the independent architecture solution, synchronous relationship can be easily found out. While, the simple network administration solution is more suitable for large-scale applications of network synchronization, such as multi-area stock trading system,

telecom service system or SCADA of large power station, in which higher accuracy time can be achieved when the solution is adopted. Time services based on computer network, compared with traditional time services offered by telephone, radio or television, provides more extensive services and consequently has a promising future.

6. REFERENCES

- [1] D.L.Mills. "Network Time Protocol Specification, Implementation and Analysis". DARPA Netw Group Report RFC-1305, Univ. Delaware, 1992
- [2] He Peng, Wu Haitao. "Study of Time Synchronization in Distributed System". DCABES2001 PROCEEDINGS, Hubei: Science and Technology Press 2001(10), pp196-198
- [3] He Peng, Li Jing. "Study and Implementation of Network Time Synchronization Algorithms". Computer Application, 2003(2), pp 15-17
- [4] He Peng, Zeng Weinu. "Study and Programming of Time Synchronization in SCADA System". Computer Engineering, 2000(7), pp 47-50
- [5] [He Peng, Zhou Jiuyan. "Design and Application of Synchronization Device in Distributed Control System". Computer Engineering, 2003(7), pp 41-43
- [6] He Peng. "Implementation of Time Synchronization of DCS with GPS", Journal of Three Gorges University, 2001, (1), pp.44-47
- [7] Michael Lombardi. "Computer Time Synchronization", [Http://www.bould.nist.gov](http://www.bould.nist.gov)



He Peng is a professor in College of Electrical and Information and a head of Information Technology Center, Three Gorges University. He graduated from Hefei Industrial University with specialty of computer application in 1986; from Xi'an Jiaotong University with specialty of software in 1989. He worked in National Time-service Center of CAS; participated in

almost 20 research projects, including the seventh national 5-year-plan, the rehearsal of 'eight-five' project held by State Bureau of Surveying and Mapping, CAS youth fund project, Hubei technology research-program, etc. He ever won the western young scientist's achievement award and the third class award of technology advancement held by CAS. Over 30 journal papers have been published, some of which were embodied by ISTP. His researches focus on active time synchronization within distributed system and Internet.

A Distributed Computing Platform ----BOINC*

Fan Yang¹, Xin-Zhong Zhu², Jian-Min Zhao¹

¹College of Computer Science and Engineering, Zhejiang Normal University

²Institute of Computer Science Studies, Zhejiang Normal University
Jinhua, Zhejiang 321004, China

Email: xinzhong@mail.zjnu.net.cn Tel: (86) 05792282145

ABSTRACT

In recent years, distributed computing has become an increasingly popular source of computing power. And more and more people have known that distributed computing is a science which solves a large problem by giving small parts of the problem to many computers to solve and then combining the solutions for the parts into a solution for the problem. What's more, recent distributed computing projects have been designed to use the computers of hundreds of thousands of volunteers all over the world, via the Internet, to look for extra-terrestrial radio signals, to look for prime numbers so large that they have more than ten million digits, and to find more effective drugs to fight the AIDS virus. These projects are so large, and require so much computing power to solve, that they would be impossible for any one computer or person to solve in a reasonable amount of time.

However computing Platforms are software client applications that we can run on our computers and that host various, often unrelated, project applications. As a distributed computing platform, Berkeley Open Infrastructure for Network Computing (BOINC) is a software platform for projects, like distributed.net and SETI@home (SETI, Search for Extraterrestrial Intelligence), that use millions of volunteer computers as a parallel supercomputer." Source code is available for the platform, and interested C++ developers are encouraged to help develop the platform code.

Keywords: Distributed Computing, BOINC, SETI

1. INTRODUCTION

Currently there are a few notable distributed computing platforms such as SETI@home [1], distributed.net [3], and United Devices [5]. They work on the principle of a user donating their machine to the system so that its free resources can help to process computationally large problems.

The widespread success of the Internet has meant that these distributed systems have been able to harness a huge amount of computational resources from the donors' machines, which would otherwise have not been utilized to their full potential. These distributed systems rely on a client-server model, where the distributed system has one server and many clients. In practice it has been used by the SETI@home distributed system very successfully, with up to three million client machines as part of the system [1].

However, since there is only one server (single machine or cluster) for all of the clients there is thus a finite limit on the number of clients the system can handle at any one time, with this limit depending on the network resources and

computational resources at the server. A common solution is to increase the bandwidth of the server's Internet connection and to upgrade the power of the server, but this can be expensive. The Berkley Open Infrastructure Networking Computing (BOINC) [2] is a programmable successor to SETI@home.

BOINC makes it fairly easy and cheap to convert an existing application to a public computing project. And BOINC projects are autonomous; each one maintains its own servers and databases, and does not depend on others. Participants can register with multiple projects, and can control how their resources are shared (for example, a user might devote 60% of his CPU time to studying global warming, and 40% to SETI). And BOINC can also be regarded as a complement to Grid systems that support resource sharing within and among institutions but do not support public computing.

In Sect. 2, we introduce the BOINC model. Implementation and the data BOINC projects export are discussed in Sect. 3, and we conclude in Sect. 4.

2. OVERVIEW OF THE SYSTEM

BOINC was originally developed to support SETI@home. However, other distributed computing projects may use BOINC. BOINC allows anyone to participate in multiple projects, and to control how his (or her) resources are divided among these projects.

And the projects of BOINC are independent, and each maintains its own servers. The BOINC developers and the University of California have no control over the creation of BOINC-based projects, and in general do not endorse them. Meanwhile, the BOINC core client is available for the following platforms:

- Windows (98 and up)
- Linux (on X86 and perhaps others)
- Solaris/SPARC
- Mac OS X

There are no specific hardware requirements (CPU speed, RAM, disk space, etc.). However, these factors may limit the amount or type of work that is sent to client's computer. Each 'work unit' has minimum RAM and disk requirements, and a deadline for completion of its computation. A BOINC project won't send a work unit to a computer that can't handle it.

If software is not available for client's computer, you may still be able to participate in BOINC projects if you are able to compile the software yourself.

* This work is supported by ZSFC research project: ZD0108

2.1 BOINC's features

BOINC is a software platform for distributed computing using volunteer computer resources. The BOINC's features fall into several areas:

a) Resources are shared among the independent projects. That is to say that many different projects can use BOINC. Projects are independent; each one operates its own servers and databases. However, projects can share resources in the following sense: Participants install a core client program, which in turn downloads and executes project-specific application programs. Participants control which projects they participate in, and how their resources are divided among these projects. When a project is down or has no work, the resources of its participants are divided among the other projects in which the participants are registered.

b) BOINC provides features that simplify the creation and operation of distributed computing projects. So the objects of BOINC have the following features:

- **Flexible application framework**

Existing applications in common languages (C, C++, Fortran) can run as BOINC applications with little or no modification. An application can consist of several files (e.g. multiple programs and a coordinating script). New versions of applications can be deployed with no participant involvement.

- **Security**

BOINC protects against several types of attacks. For example, it uses digital signatures based on public-key encryption to protect against the distribution of viruses.

- **Multiple servers and fault-tolerance**

Projects can have separate scheduling and data servers, with multiple servers of each type. Clients automatically try alternate servers; if all servers are down, clients do exponential backoffs to avoid flooding the servers when they come back up.

- **System monitoring tools**

BOINC includes a web-based system for displaying time-varying measurements (CPU load, network traffic, database table sizes). This simplifies the task of diagnosing performance problems.

- **Source code availability**

BOINC is distributed under a public license that allows it to be used freely for public or private distributed computing projects, with the restriction that it cannot be used as the basis for commercial products. BOINC applications need not be open source. Each project must provide and maintain its own server systems; these systems can be set up easily using open-source components (MySQL, PHP, Apache).

- **Supports for large data**

BOINC supports applications that produce or consume large amounts of data, or that use large amounts of memory. Data distribution and collection can be spread across many servers, and participant hosts transfer large data unobtrusively. Users can specify limits on disk usage and network bandwidth. Work is dispatched only to hosts able to handle it.

c) BOINC provides the following features to participants:

- **Multiple participant platforms**

The BOINC core client is available for most common platforms (Mac OS X, Windows, Linux and other Unix systems). The client can use multiple CPUs.

- **Web-based participant interfaces**

BOINC provides web-based interfaces for account creation, preference editing, and participant status display. A participant's preferences are automatically propagated to all their hosts, making it easy to manage large numbers of hosts.

- **Configurable host work caching**

The core client downloads enough work to keep its host busy for a user-specifiable amount of time. This can be used to decrease the frequency of connections or to allow the host to keep working during project downtime.

2.2 Administrative web interface

Normally we don't have to directly examine or manipulate the BOINC database. If we need to, we can use the MySQL command-line interpreter or BOINC's administrative web interface. And BOINC's administrative web interface provides interfaces for:

- Browsing the database
- Screening user profiles
- Viewing recent results
- Browsing strip charts
- Browsing log files
- Creating user accounts

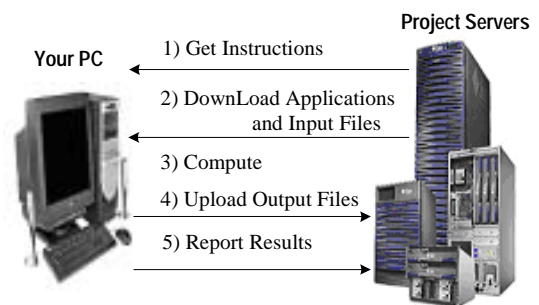
3. IMPLEMENTATION

3.1 Implementation of BOINC

It's easy to participate in a BOINC project.

- Create an account .Go to the project's web site, click on create account, and fill out the form. You will receive an email containing your account ID (a long random string). Save this email.
- Download and install BOINC. Download BOINC for your type of computer, install it, and run it. You will be asked to enter the project's URL and your account ID.

The implementation of BOINC is as follows:



- (1). Your PC gets a set of instructions from the project's scheduling server. The instructions depend on your PC: for example, the server won't give it work that requires more RAM than you have. The instructions may include many multiple pieces of work. Projects can support several applications, and the server may send you work from any of them.
- (2). Your PC downloads executable and input files from the project's data server. If the project releases new versions of its applications, the executable files are downloaded automatically to your PC.

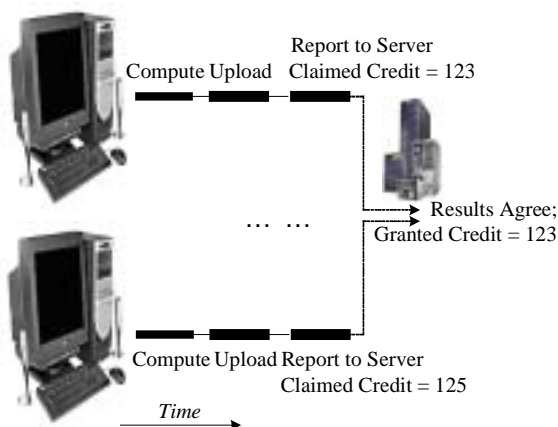
- (3). Your PC runs the application programs, producing output files.
- (4). Your PC uploads the output files to the data server.
- (5). Later (up to several days later, depending on your work buffer preferences) your PC reports the completed results to the scheduling server, and gets instructions for more work.

This cycle is repeated indefinitely. BOINC does this all automatically; you don't have to do anything.

3.2 Credit

The project's server keeps track of how much work your computer has done; this is called credit. To ensure that credit is granted fairly, BOINC works as follows:

- Each work unit may be sent to several computers.
- When a computer reports a result, it claims a certain amount of credit, based on how much CPU time was used.
- When at least two results have been returned, the server compares them. If the results agree, then users are granted the smaller of the claimed credits.



Please keep in mind:

- There may be a delay of several days between when your computer reports a result and when it is granted credit for the result. Your User page shows you how much credit is 'pending' (claimed but not granted).
- The credit-granting process starts when your computer reports a result to the server (not when it finishes computing the result or uploading the output files).
- In rare cases (e.g. if errors occur on one or more computers) you may never receive credit for a computation.

4. THE DATA BOINC PROJECTS EXPORT

BOINC projects may export data describing teams, users and hosts. This data is exported in XML files that can be downloaded via HTTP.

And the data is presented in several different 'views': teams ordered by credit, teams ordered by ID, etc. Each view is available in two ways:

- As a single file.
- Broken into a number of files, each containing a fixed number of records. This lets client get a single record or range of records efficiently.

The files are as follows:

Tables.xml

For each table (team, user, and host) this gives

- The total number of records
- The number of records per file for summary files
- The number of records per file for detail files

It also includes the UNIX time when the files were last generated, and a list of the project's applications, with counts of results in various states.

For example:

```
<tables>
  <update_time>1046220857</update_time>
  <nusers_total>127</nusers_total>
  <nusers_per_file_summary>1000</nusers_per_file_summary>
  <nusers_per_file_detail>100</nusers_per_file_detail>
  <nteams_total>14</nteams_total>
  <nteams_per_file_summary>1000</nteams_per_file_summary>
  <nteams_per_file_detail>100</nteams_per_file_detail>
  <nhosts_total>681</nhosts_total>
  <nhosts_per_file_summary>1000</nhosts_per_file_summary>
  <nhosts_per_file_detail>100</nhosts_per_file_detail>
  <applications>
    <application>
      <name>setiathome</name>
      <results_unsent>100</results_unsent>
    </application>
    ... ..
  </applications>
</tables>
```

team_total_credit.xml, team_total_credit_N.xml

Team summaries, ordered by decreasing total credit. N is 0, 1, ...

team_expavg_credit.xml, team_expavg_credit_N.xml

Team summaries, ordered by decreasing recent-average credit.

team_id.xml, team_id_N.xml

Team details, ordered by increasing ID.

user_total_credit.xml, user_total_credit_N.xml

User summaries, ordered by decreasing total credit.

user_expavg_credit.xml, user_expavg_credit_N.xml

User summaries, ordered by decreasing recent-average credit.

user_id.xml, user_id_N.xml

User details, ordered by increasing ID.

host_total_credit.xml, host_total_credit_N.xml

Host summaries, ordered by decreasing total credit.

host_expavg_credit.xml, host_expavg_credit_N.xml

Host summaries, ordered by decreasing recent-average credit.

host_id.xml, host_id_N.xml

Host details, ordered by increasing ID.

The format of the various XML elements is as follows:

Team summary

```
<team>
  <id>5</id>
  <name>Broadband Reports Team Starfire</name>
  <total_credit>153402.872429</total_credit>
  <expavg_credit>503030.483254</expavg_credit>
  <nusers>14</nusers>
</team>
```

Team detail

```
<team>
  <id>5</id>
```

```

<name>Broadband Reports Team Starfire</name>
<total_credit>153402.872429</total_credit>
<expavg_credit>503030.483254</expavg_credit>
<nusers>14</nusers>
<create_time>0</create_time>
<name_html>%3Ca%20href%3D%27http%3A%2F%2Fbroadb
andreports%2Ecom%2Fforum%2Fseti%2
7%3E%3Cimg%20src%3D%27http%3A%2F%2Fi%2Edslr%2
Enet%2Fpics%2Ffaqs%2Fimage2067%2Ejpg%27%3E</name_html>

```

```

<country>None</country>
<user>
  <id>12</id>
  <name>John Keck</name>
  <total_credit>42698.813543</total_credit>
  <expavg_credit>117348.653646</expavg_credit>
  <teamid>5</teamid>
</user>
<user>
  <id>14</id>
  <name>Liontaur</name>
  <total_credit>46389.595430</total_credit>
  <expavg_credit>122936.372641</expavg_credit>
  <teamid>5</teamid>
</user>
</team>

```

User summary

```

<user>
  <id>12</id>
  <name>John Keck</name>
  <total_credit>42698.813543</total_credit>
  <expavg_credit>117348.653646</expavg_credit>
  [ <teamid>5</teamid> ]
  [ <has_profile/> ]
</user>

```

User detail

```

<user>
  <id>3</id>
  <name>Eric Heien</name>
  <total_credit>4897.904591</total_credit>
  <expavg_credit>9820.631754</expavg_credit>
  <country>United States</country>
  <create_time>1046220857</create_time>
  [ <teamid>14</teamid> ]
  [ <has_profile/> ]
</user>
<host>
  <id>27</id>
  <total_credit>0.000000</total_credit>
  <expavg_credit>0.000000</expavg_credit>
  <p_vendor></p_vendor>
  <p_model></p_model>
  <os_name>Darwin</os_name>
  <os_version>6.2</os_version>
</host>

```

```

<host>
  <id>266</id>
  <total_credit>0.000000</total_credit>
  <expavg_credit>0.000000</expavg_credit>
  <p_vendor>GenuineIntel</p_vendor>
  <p_model>Intel(R)</p_model>
  <os_name>Linux</os_name>
  <os_version>2.4.18-18.7.x</os_version>
</host>
</user>

```

Host summary

```

<host>

```

```

  <id>266</id>
  <total_credit>0.000000</total_credit>
  <expavg_credit>0.000000</expavg_credit>
  <p_vendor>GenuineIntel</p_vendor>
  <p_model>Intel(R)</p_model>
  <os_name>Linux</os_name>
  <os_version>2.4.18-18.7.x</os_version>
</host>

```

Host detail

```

<host>
  <id>102</id>
  <userid>3</userid>
  <total_credit>0.000000</total_credit>
  <expavg_credit>0.000000</expavg_credit>
  <p_vendor>GenuineIntel</p_vendor>
  <p_model>Pentium</p_model>
  <os_name>Windows XP</os_name>
  <os_version>5.1</os_version>
  <create_time>1040170006</create_time>
  <timezone>28800</timezone>
  <ncpus>2</ncpus>
  <p_flops>45724737.082762</p_flops>
  <p_iops>43233895.373973</p_iops>
  <p_membw>4032258.064516</p_membw>
  <m_nbytes>670478336.000000</m_nbytes>
  <m_cache>1000000.000000</m_cache>
  <m_swap>1638260736.000000</m_swap>
  <d_total>9088008192.000000</d_total>
  <d_free>3788505088.000000</d_free>
  <n_bwup>24109.794088</n_bwup>
  <n_bwdwn>57037.049858</n_bwdwn>
</host>

```

5. CONCLUSIONS

In distributed computing, many peoples have sawn the potential to revolutionize the way science is conducted around the world. Several months ago, BOINC was just in limited-user beta testing, but now a version customized for SETI@home is being set to enter public usages. We all believe that in the near future, a web site will allow home users to pick and choose amongst the various BOINC-based projects, downloading specific applications that pertain to the research that most interests them and customizing how much of their PC resources they want to make available.

6. REFERENCES

- [1] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Massively distributed computing for SETI. Computing in Science & Engineering, 3(1):78–83, Feb. 2001.
- [2] <http://boinc.berkeley.edu>.
- [3] <http://www.distributed.net>.
- [4] Haluk Topcuoglu, Salim Hariri and Min-you Wu . Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing. IEEE Transactions on Parallels and Distributed Systems, 13:260–274, Mar. 2002.
- [5] <http://www.ud.com>.

The Application of ACE in The Distributed Network Management System

¹Chen Jun ²Guo Qingping

School of Computer Science and Technology, Wuhan University of Technology, Hubei 430063, China

¹ cjwuhan@sina.com

Tel: +86 13618655728

² qpguo@mail.whut.edu.cn

Tel: +86 (0)27 86554639

ABSTRACT

We will discuss how to make use of the power of ACE (Adaptive Communication Environment) to develop a distributed network management system.

Keywords: ACE, network management, snmp

1. INTRODUCTION

Computing power and network bandwidth have increased dramatically over the past decade. However, the design and implementation of complex software remains expensive and error-prone. Much of the cost and effort stems from the continuous re-discovery and re-invention of core concepts and components across the software industry. In particular, the growing heterogeneity of hardware architectures and diversity of operating system and communication platforms makes it hard to build correct, portable, efficient, and inexpensive applications from scratch.

Object-oriented (OO) application frameworks have proved to be a mature technology to reduce the cost and improve the quality of software. The primary benefits of OO application frameworks stem from the modularity, reusability, extensibility, and inversion of control to developers.

The Adaptive Communication Environment (ACE) is a widely-used, open-source OO frameworks written in C++ that implements core concurrency and networking patterns for communication software. It is the backbone to realize our distributed network management system called BwN2000.

2. THE ACE ARCHITECTURE

ACE has a layered design, with the following three basic layers in its architecture:

2.1 The Operating System (OS) Adaptation Layer

The OS Adaptation is a thin layer of C++ code that sits between the native OS APIs and the rest of ACE. This layer shields the higher layers of ACE from platform dependencies, which makes code written with ACE relatively platform independent. Thus, with little or no effort developers can move an ACE application from platform to platform.

2.2 The C++ Wrapper Façade Layer

The C++ wrapper façade layer includes C++ classes that can be used to build highly portable and typesafe C++ applications. This is the largest part of the ACE toolkit and includes approximately 50% of the total source code. C++ wrapper classes are available for:

- 1) Concurrency and synchronization

- 2) Memory management components
- 3) Timer classes
- 4) Container classes
- 5) Signal handling
- 6) Filesystem components
- 7) Thread management

2.3 The Frameworks and Patterns Layer

The ACE framework components are the highest-level building blocks available in ACE. These framework components are based on several design patterns specific to the communication software domain. A designer can use these framework components to build systems at a much higher level than the native OS API calls. These framework components are therefore not only useful in the implementation stage of development, but also at the design stage, since they provide a set of micro-architectures and pattern languages for the system being built.

3. NETWORK MANAGEMENT AND SNMP

Requirements of Network Management

OSI Management is required for a number of purposes. These requirements are categorized into a number of functional areas.[1]

- fault management
- accounting management
- configuration management
- performance management
- security management

The Simple Network Management Protocol

SNMP is the most popular protocol used to manage networked devices. It was designed in the late 1980s to facilitate the exchange of management information between networked devices operating at the application layer of the ISO/OSI model. SNMP is formally defined in RFC 1157[2]:

Implicit in the SNMP architectural model is a collection of network management stations and network elements. Network management stations execute management applications which monitor and control network elements. Network elements are devices such as hosts, gateways, terminal servers, and the like, which have management agents responsible for performing the network management functions requested by the network management stations. The Simple Network Management Protocol (SNMP) is used to communicate management information between the network management stations and the agents in the network elements.

An SNMP-managed network typically consists of three components: managed devices, agents, and one or more network management systems.

A managed device can be any piece of equipment that sits on

your data network and is SNMP compliant. Routers, switches, hubs, workstations, and printers are all examples of managed devices.

An agent is typically software that resides on a managed device. The agent collects data from the managed device and translates that information into a format that can be passed over the network using SNMP.

A network-management system monitors and controls managed devices. The network management system issues requests, and devices return responses.

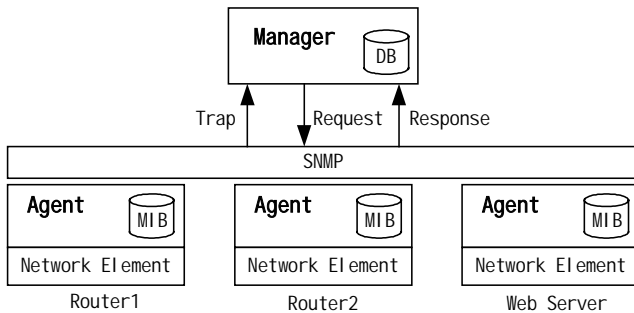


Fig 1 Centralized Network Management Architecture

4. THE BwN2000 ARCHITECTURE

Requirements

a) Distributed.

The centralized network management is not suitable to A client could manage several network elements. A network element could be managed by several clients.

b) Cross-platform.

The program should run under Windows2000 and Redhat Linux 8.0 plus. The code is the same under two different environments except that the Makefile is not.

c) SNMP based.

Architecture

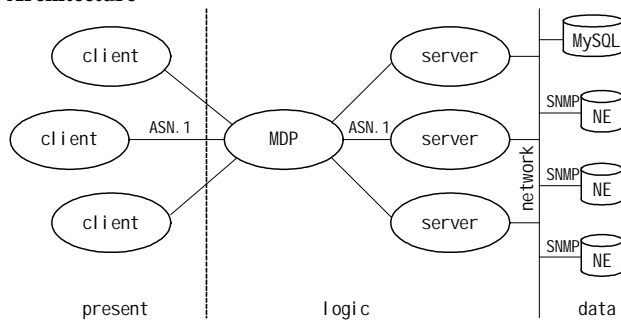


Fig 2 Three-tier architecture

the client, server, MDP are all processes. The client is responsible for the presentation of data received from servers and sends command messages to manipulate the network elements (NE). The server receives command messages from clients or trap messages from NEs and manage NEs with SNMP. Messages between the client and the server are dispatched by MDP. MDP is short for “message dispatch process”, which is essential to realize the distributed architecture in BwN2000.

Messages between clients and servers are encoded in ASN.1 format.

5. PATTERNS AND FRAMEWORKS

Patterns & Frameworks

“Design patterns are not about designs such as linked lists and hash tables that can be encoded in classes and reused as it. Nor are they complex, domain-specific designs for an entire application or subsystem. The design patterns in this book are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.[3]” And frameworks can be viewed as a concrete reification of families of design patterns that are targeted for a particular application-domain.

Reactor Pattern[4]

The Reactor pattern allows event-driven applications to demultiplex and dispatch service requests that are delivered to an application from one or more clients. The structure introduced by the Reactor pattern ‘inverts’ the flow of control with an application, which is known as the Hollywood Principle “Don’t calls us, we’ll call you”

The Reactor in ACE works in conjunction with several components. The basic idea is that the Reactor framework determines that an event has occurred and issues a “callback” to a method in a pre-registered event handler object. This object is implemented by the application developer and contains application specific code to handle the event.

```
class ACE_Event_Handler
{
public:
    // called when input events occur (e.g., connection or data).
    // the programmer should override this method
    virtual int handle_input(ACE_HANDLE);
    // Get the I/O handle.
    virtual ACE_HANDLE get_handle();
    virtual ~ACE_Event_Handler();
    ...
};

class ACE_Reactor
{
public:
    // Register handler for OS events.
    int register_handler(ACE_Event_Handler *event_handler,
                       ACE_Reactor_Mask mask);
    int handle_events();
    ...
}
```

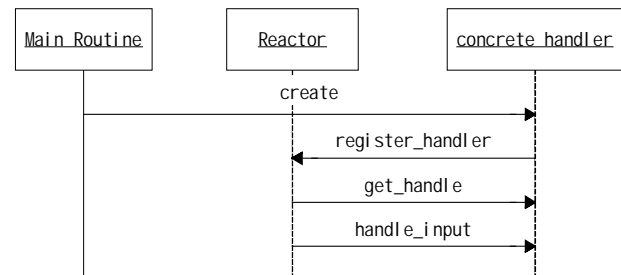


Fig 3 Reactor Pattern

This pattern is easy to use. The user (the application developer) will

- 1) Create an Event Handler to handle an event he is interested in.
- 2) Register with the Reactor, informing it that he is interested in handling an event

The Reactor framework then automatically will

- 1) The Reactor maintains tables internally, which associate different event types with event handler objects
- 2) When an event occurs that the user has registered for, it issues a call back to the appropriate method in the handler.

IPC SAP (Interprocess Communication Service Access Pointer wrappers)

Sockets, TLI, STREAM pipes and FIFO's provide a wide range of interfaces for accessing both local and global IPC mechanisms. However, there are many problems associated with these none-uniform interfaces. Problems such as lack of type safety and multiple dimensions of complexity lead to problematic and error-prone programming.

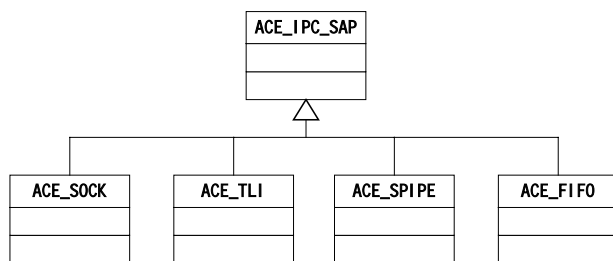


Fig 4 IPC SAP

The IPC SAP classes are divided into four major categories based on the different underlying IPC interface they are using. The class diagram above illustrates this division. For example, The *ACE_SOCKET* class contains functions that are common to the BSD sockets programming interface. Underneath each of these four classes lies a whole hierarchy of wrapper classes that completely wrap the underlying interface and provide highly reusable, modular, safe and easy-to-use wrapper classes.

Three classes under the *ACE_SOCKET* are presented below.

ACE_SOCKET_Acceptor is used for passive connection establishment based on the BSD *accept()* and *listen()* calls.

ACE_SOCKET_Connector is used for active connection establishment based on the BSD *connect()* call.

ACE_SOCKET_Stream is used to provide TCP-based connection-oriented messaging service.

ACE_Thread & ACE_Thread_Manager

There are several different interfaces that are available for thread management on different platforms. These include the POSIX pthreads interface, Solaris threads, Win32 threads etc. Each of these interfaces provides the same or similar functionality but with APIs that are vastly different. This leads to difficult, tedious and error-prone programming, since the application programmer must make himself familiar with several interfaces to write on different platforms. Furthermore, such programs, once written, are non-portable and inflexible. *ACE_Thread* provides a simple wrapper around the OS thread calls that deal with issues such as creation, suspension,

cancellation and deletion of threads. This gives the application programmer a simple and easy-to-use interface which is portable across different threading APIs. *ACE_Thread* is a very thin wrapper, with minimal overhead. Most methods are inclined and thus are equivalent to a direct call to the underlying OS-specific threads interface. All methods in *ACE_Thread* are static and the class is not meant to be instantiated.

The threads are created by using the *ACE_Thread::spawn()* or *ACE_Thread::spawn_n()* call. This call is passed a pointer to the function which is to be called as the starting point of execution for the thread.

The *ACE_Thread_Manager* provides a superset of the facilities that are available in *ACE_Thread*. In particular, it adds management functionality to make it easier to start, cancel, suspend and resume a group of related threads.

ACE_Task

ACE_Task can be used as a Higher Level Thread (which we can call a *Task*) or as an Active Object in the Active Object Pattern.

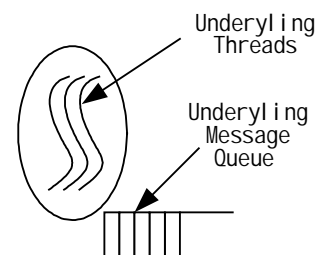


Fig 5 ACE_Task

The above diagram shows that each *Task* contains one or more threads and an underlying message queue. *Tasks* communicate with each other through these message queues. However, the message queues are not entities that the programmer needs to be aware of. A sending *Task* can just use the *putq()* call to insert a message into the message queue of another *Task*. The receiving *Task* can then extract this message from its own message queue by using the *getq()* call. Such an architecture helps considerably in simplifying the programming model for multi-threaded programs.

When a process (client or server) connects to MDP, A thread is spawned in MDP to accept messages from the process and a *Task* is created at the same time to wait for messages to be sent to the process.

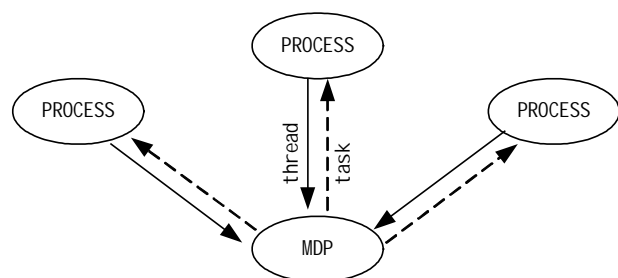


Fig 6 MDP communication model

The relevant section of the code is reproduced below.

The main routine in MDP is simple, Reactor Pattern is used.

```
int main(int argc, char * argv[])
{
    ACE_INET_Addr addr(PORT_NO);
    ... ..
    // class Bw_Accept derives from ACE_SOCK_Acceptor
    Bw_Accept_Handler *eh=new Bw_Accept_Handler(addr);
    ACE_Reactor::instance()->register_handler(eh,
        ACE_Event_Handler::ACCEPT_MASK);
    while(1)
        ACE_Reactor::instance()->handle_events();
}
```

A Task and a thread are created in the handler function.

```
int Bw_Accept_Handler::handle_input(ACE_HANDLE handle)
{
    ACE_SOCK_Stream *pStream=new ACE_SOCK_Stream;
    If (this->peer_acceptor.accept(*pStream, // stream
        0, // remote address
        0, // timeout
        1) == -1) // restart if interrupted
    {
        ACE_DEBUG((LM_ERROR,"Error in connection\n"));
        return 0;
    }
    // class Waiter derives from ACE_Task
    Waiter *pWaiter = new Waiter(pStream);
    ACE_Thread_Manager::instance()->
        spawn((ACE_THR_FUNC)WorkThread,(void *)pWaiter);
    return 0;
}
```

6. CONCLUSION

ACE is an effective communication framework. It is more simple to realize a distributed architecture than CORBA or socket APIs. And the convenience does not sacrifice the performance. In our projects, facts prove that ACE help to reduce the development duration, increase the robustness of our codes. Frank Buschmann says that Even if the heavy-weight middleware declines, ACE will still be preferable to our needs[5]. I completely agree with him in this point.

7. REFERENCES

- [1] ISO/IEC 7489-4. 1989-11-15.
- [2] Case J, Fedor M, Schoffstall M, et al. RFC1157: A Simple Network Management Protocol (SNMP). 1990-05-01.
- [3] Erich Gamma, Richard Helm, et al. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995. 2 - 3
- [4] Umar Syid. A Tutorial Introduction to the ADAPTIVE Communication Environment (ACE). <http://www.cs.wustl.edu/~schmidt/PDF/ACE-tutorial.pdf>. 53 - 72
- [5] Douglas C.Schmidt, Stephen D.Huston. C++ Network Programming, Volume2 Systematic Reuse with ACE and Frameworks. Addison-Wesley, 2003.



ChenJun (1976~) is a master candidate in school of Computer Science and Technology, Wuhan University of Technology. His research interests are in network management, wbm etc.

Guo Qingping is a Full Professor and a head of Parallel Processing Lab, dean of Computer Technology Institute in School of Computer Science and Technology, Wuhan University of Technology. He is one of the DCABES international conference founder, was the chairman of DCABES 2001, co-chair of DCABES 2002, and the chairman of DCABES 2004.

A Link Layer Automatic Topology Discovery Algorithm

Liu Yuhua^{1,2}, Yu Shengsheng¹, Li Yanhong², Zhang Xiaopeng²

¹ College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China 430074

² Department of Computer Science, Central China Normal University, Wuhan, China 430079

Email: yhliu@mail.ccnu.edu.cn Tel: +86 (0) 27-62265962

ABSTRACT

After analyzing the importance and current status of a link layer topology discovery, it is discussed that the theory, model and principle of topology discovery technology are based on Bridge MIB' AFT. In this paper a general algorithm of a link layer topology discovery based on AFT is presented, its pseudo code along with the detailed description is given out also. At last a conclusion of a link layer topology discovery algorithm is summarized.

Keywords: A link layer topology, switch domain, switch, discovery algorithm

1. INTRODUCTION

With the development of automatic topology discovery technology for IP networks, the network layer discovering technology is almost mature. To realize a layer-3 topology discovery is relatively easy compared with layer-2. The main elements of network layer are routers, logical net-segments and subnets. In order to perform routers' basic routing function, routing tables of routers learn to obtain the whole routing distributing information of a network^[1]. Thus a router can apperceive its near router and the subnet it connects. However, layer-3 topology discovery covers only a small fraction of elements and interconnections in an IP network and it fails to obtain the complex relationships among elements of link layer (switchs, bridges, hubs and routers without SNMP agent) that compose a Ethernet LAN. Nowadays the popular Ethernet technology is very link layer technology. More and more switches are distributed in Ethernet to provide wider bandwidth through subnet tiny division. This kind of network structure seems to keep increasing. Under this condition, the traditional network management becomes insufficient on peer-to-peer link control, underlying conflicts management and device failure management. To improve the network efficiency, discovering the distribution of link layer physical devices and their connections is the precondition of modern network management.

Unfortunately, there's no perfectly mature method for layer-2 topology discovery. The inherent transparency of layer-2 hardware poses several difficult challenges to its topology discovery. Hardware providers, such as Cisco and Intel, have designed topology discovery protocols and tools for their own products, but these tools are of no use in heterogeneous multi-vendor environment. Recently, the professional workers have acknowledged the importance of physical topology. IETF has instituted RFC2922 protocol standard^[2] and designed a Physical Topology SNMP Management Information Base. Some other experts are trying to design a special link layer discovery protocol LLDP^[3]. But these standards and protocols are not perfect enough to essentially work. On base of link layer topology discovery research^[4,5,6], we present in this paper a SNMP-based algorithm of data link layer topology

discovery for heterogeneous IP network based on SNMP protocol which is widely supported by network devices and AFT (Address Forward Table) in switch's Bridge MIB, which is able to solve the problem for link layer topology discovery.

2. THEORY AND MODEL

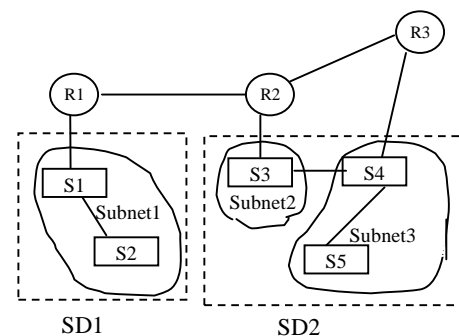
2.1 Concepts and Model

Definition 1 A computer network can be figured as an undirected graph $G=(V, E)$; V denotes set of network nodes and E denotes set of edges between nodes. Node v represents a network node and edge e represents a physical connection between two nodes in G . $v \in V, e \in E$.

Definition 2 Subnet N is a maximal set of IP addresses (representing network elements) in G , where any two nodes in N can communicate directly with each other on layer-3 or above without involving a router, while communication across different subnets must go through a router.

Backbone of an IP network comprises router-to-router, router-to-subnet structures. Routers divide IP network to several subnets or logical segments in order to prevent network "broadcast storm". A subnet always comprises switch-switch and switch-host structures and this structure might bring link loop and subnet "broadcast storm". To avoid this, the Spanning Tree Protocol (STP) is adopted among switches to block some ports to cut loop among switches. In such case, only the switch ports on the spanning tree are allowed to transfer data frame.

Definition 3 For a maximal set S of switches which runs spanning tree protocol, if there exists a path connecting two switches and the nodes it passes by are all in S , we call such switch set S as "switch domain".



[]- switch domain ○-subnet ○-router □- switch

Figure 1 The example of the network G in the managing area

In distributed system, the set of managing objects is called management domain. So the network G is such a management domain. Every network element in management domain is identified with a unique IP address and a subnet mask that

defines the IP address space corresponding to the element's subnet. Figure 1 shows a management domain (network G), router $R1$, $R2$ and $R3$ connect one another and respectively connects to subnet1, subnet2 and subnet3. Their discovery belongs to layer-3 backbone discovery. Furthermore, because subnet1 runs a single *STP* instance, switch domain1 composes with a single subnet; Switch domain2 contains subnet2 and subnet3 and runs another *STP* instance, which is a multi-subnet topology structure. Typically, in subnet, switch $S4$ and $S5$ can communicate directly with each other. Moreover, communication between $S3$ and $S4$, although there is a direct physical connection between them, must go through $R2$ and $R3$ because they are in different subnets. A switch domain may contain one subnet N or several different subnets

$N_i (i=1,2, \dots, n)$. They distinguish from the difference of property that root switch chosed and manager's decision on switch domain size and cost of using *BPDU* traffic. This paper discusses the issue of link layer topology discovery in a switch domain.

2.2 Principle of Link Layer Topology Discovery

A primary function of a switch is automatically learning to capture source MAC addresses, where data frame comes from through its each port and storing the source MAC address with corresponding port-id into AFT which are the physical addresses of switches, bridges or other network devices connected to itself. Switches use the MAC addresses in AFT as destination to forward or filter frames. In our algorithm using the MAC addresses it starts at a root-switch in the switch domain and find out one by one all the interconnections among switches using width-first searching method. Switches in a subnet can preprocess to learn complete MAC information while switches in different subnets whose communication must go through a router can merely learn incomplete MAC information although they are directly connected. In this case, in the switch domain the condition to judge whether there's a direct connection between switches in a subnet or not is different.

Definition 4 In a switch domain we use the notation S_{ij} to identify the j th port of switch S_i and A_{ij} to denote the set AFT entries at port S_{ij} (i.e., the set of MAC addresses that have been seen as source addresses on frames received at S_{ij}). Intuitively, A_{ij} (j ranges from 1 to n) stands for the collection of MAC addresses of S_i . Notation n denotes the ports number of S_i .

Theorem 1 In a same subnet, ports S_{ij} and S_{kl} are directly connected if and only if $A_{ij} \cap A_{kl} \neq \emptyset$ and $A_{ij} \cup A_{kl} = M$ (M stands for the set of all nodes's MAC addresses in the subnet).

Theorem 2 When in different subnet S_{ij} and S_{kl} are directed connected if and only if $A_{ij} \cap A_{kl} = \emptyset$ and A_{ij} contains MAC addresses of S_k .

In link layer a switch may connect to routers and hosts other than directly connect to a switch. Especially, some desktop switches are usually at leaf node of a spanning tree and connects all hosts to desktop.

2.3 Information Collection for Link Layer Topology Discovery

We are mainly about to use the RFC-1213 defined MIB- and RFC-1443 defined Bridge MIB. IP group and System group in MIB- is to be used. The table ipNetToMediaTable

in IP group is a forward table of IP address. It maps IP address ipNetToMediaNetAddress of device to its MAC address ipNetToMediaPhysAddress. This map is used to find out devices in a subnet when doing information collecting. The value of index item ipNetToMediaIfIndex in the table equals to that of the index item in table ipAddrTable. So we can search the subnet mask ipAdEntNetMask of the index item value in table ipAddrTable to compute the ID of the subnet where the devices are in. What's more, the variable SysServices in System group that indicates which layer of ISO model the device is working in is used together with variable ipForwarding in IP group to judge the device's type. If values of the variable dot1dStpDesignateRoot in dot1dStp group of Bridge MIB of searched switches are equal, according as the rules of STP the switches are in the same switch domain.

Table1 The Address Forward Table of a switch (dot1dTpFdbTable)

dot1dTpFdbAddress	dot1dTpFdbPort	dot1dTpFdbStatus
00 04 28 75 22 1a	13	learned(3)
00 05 32 42 09 c1	21	self(4)
00 04 28 75 22 1b	14	learned(3)
00 05 01 c9 97 ff	13	learned(3)

The information in dot1dBase group and dot1dTP group of Bridge MIB is the most important for link layer discovery. In dot1dBase group, the MAC address of switch is defined with object dot1dBaseBridgeAddress and the number of switch ports is denoted by dot1dBaseNumports. The projection between dot1dBaseport and dot1dBasePortIfIndex can be found in table dot1dBasePortTable. In group dot1dTp, the Address Forward Table dot1dTpFdbTable (AFT, referred to Table 1) contains destination MAC address dot1dTpFdbAddress, corresponding port ID dot1dTpdbPort and status information dot1dTpFdbStatus which indicates whether the MAC address of the port is of itself or the destination address learned. The data frames are forward to the destination notes by the port. In table dot1dTpFdbTable, the MAC addresses of S_{ij} compose the address set A_{ij} of switch S_i .

3. LINK LAYER AUTOMATIC TOPOLOGY DISCOVERY ALGORITHM

Based on the theory and models we presented above a link layer topology discovery algorithm, which first finds out switches and other devices in a subnet using the routing table of the router that connects to the subnet and then constrained in a switch domain starts at a root-switch to discover the distribution and interconnections of switches and other devices by tracing a tree supported with Theorem 1 and Theorem 2.

3.1 Method for Subnet Topology Discovery

Before link layer topology discovering, the elements of a subnet, such as switches and routers, must be found out. We have two methods to make it come true.

The first method is mainly based on ARP protocol. Any router uses ARP protocol, with dynamic refreshing mode, to store in its Cache table a mapping from elements' IP addresses to their MAC addresses in all subnets that are directly collected to it. These IP addresses can be gained one by one using the MIB variable ipNetToMediaNetAddressTable in ARP table. Then it's to do mode matching. If successfully, it proves that

element the IP address corresponded is a member of subnet. Else it doesn't. The merit of this method is gaining values fast and its shortage is that it can't do dynamic real time refreshing. Here is a reparative skill - according to round- enquiring time user set, using program by Telnet simulating to modify the time interval and to refresh the ARP table.

The second method is mainly based on ICMP protocol. The 5 steps needed to carry it out presented below:

- (1) Ascertain gateway, address and mask of the subnet. The address gained by "and" IP and mask;
- (2) Get subnet type from gateway address;
- (3) Gain the machine number by using subnet address and subnet mask;
- (4) "Ping" IP addresses in the subnet space and memorize the inspected IP address into IP address table;
- (5) According to the request of socket, other information of live machine can be gained, such as the domain name of IP address.

3.2 Link Layer Automatic Topology Discovery Algorithm

Given all elements of the subnet having been ascertained, ulteriorly topology discovery algorithm can be used to make sure the complex collections among these elements. Four queues are defined: routerList, deviceList, switchList, discoveryList. The pseudo codes of algorithm are followed.

```

routerList=FindRoutersInSubnets(D); //step1
for(each r in routerList){ //step2
    dv=retrieveMIB(r); addElement(dv,deviceList); }
for(each d in deviceList){ //step3
    if(run_agent(d) && isSwitch(d)){
        if(retrieveBridgeMIB(d, "dot1dStpDesignateRoot")=md)
        {
            makeFlag(d,"switch");addElement(d,switchList);confirmFlag("false",d,switchList);
            removeElement(d,deviceList); } }
ping_switch(switchList); //step4
for(each i in switchList) retrieveBridgeMIB(i); //step5
Mi=getMACSubnet(Ni);
clearList(discoveryList);s=findRootSwitch(switchList); //step6
confirmFlag("true",s,switchList);addElement(s,discoveryList);
drawDevice(s);
while(!empty(discoveryList)){ //step7
    Si=removeElement(discoveryList);
    for (each port j of switch Si){ //step8
        Aij=FindMACSet(j,Si);
        If (find_match(Aij,switchList,Sk)){ //step8.1
            S1=findSubnet(Si);S2=findSubnet(Sk);Akl=FindMACSet(l,Sk);
            if(S1=S2&&Aij Akl=Msi&&Aij Akl=Φ){ //step8.2
                addSSLink(Sij,Skl);
            }else if(S1!=S2&&Aij Akl=Φ){ //step8.3
                addSSLink(Sij,Skl);
            }else{continue;}
            confirmFlag("true",Sk,switchList); //step8.4
            addElement(Sk,discoveryList);drawDevice(Sk);
        }else{ //step8.5
            find_match(Aij,deviceList,h);
            addSHLink(Si,h);drawDevice(h); } }
    }
} //step9

```

3.3 Detailed description of the Algorithm

Step1. Capture the *routerList* of all routers connected to subnets in the switch domain. Notation 'D' denotes the switch domain.

Step2. For each router in *routerList*, visiting its *ipNetToMediaTable* of MIB- to get known the devices that *ipNetToMediaNetAddress*(IP address) and *ipNetToMediaPhysAddress* (MAC address) denote. Then using *ipNetToMediaIfIndex* to find out in the *ipAddrTable* of MIB- *ipAdEntNetMask* to make sure the mask of subnet the device belongs to.

Step3. For each device active and running SNMP protocol in *deviceList*, checking *sysServices* and *ipForwarding* value in its MIB to judge the type of it (a switch, a router or a host) and note it into *deviceList*. For a switch, go on to check the value of *dot1dStpDesignateRoot* in *BridgeMIB* to confirm its switch domain (*md* denotes the switch domain ID). Then add the switch into *switchList* if it's in the same switch domain which switches in *switchList* belong to and initiate "confirmFlag" with "false" before removing it from *deviceList*.

Step4. All devices in the subnet execute operates of "Ping" to each switch in *switchList* in turn in order to make ports learn complete MAC addresses.

Step5. Visiting *BridgeMIB* of each switch in *switchList* to pick up information of the Address Forward Table *dot1dTpFdbTable* (AFT), *dot1dBaseNumports* and *dot1dBasePortTable*. Symbol *M_i* denotes device's MAC address set of subnet *N_i*.

Step6. Emptying *discoverList* queue, finding out the root-switch in *switchList* and setting its confirmFlag as "true". Then adding it to *discoverList* and drawing it in topology map.

Step7. If *discoverList* is not empty, moving out a switch *S_i* to check; else go to Step9.

Step8. For each port *j* of *S_i*, getting its *A_{ij}* and,

8.1 If a MAC address in *A_{ij}* matches that of a device *S_k* in *switchList*, doing more to get MAC address collection *A_{kl}* of each port *l* of *S_k*;

(1) If *S_i* and *S_k* belong to the same subnet, ports *S_{ij}* and *S_{kl}* are directly connected if and only if *A_{ij} A_{kl}=* and *A_{ij} A_{kl}=M* (*M* expresses the set of all MAC addresses in the subnet). Then jump to (3);

(2) When *S_i* and *S_k* are in different subnet, *S_{ij}* and *S_{kl}* are directly connected if and only if *A_{ij} A_{kl}=* and *A_{ij}* contains MAC addresses of *S_k*.

(3) Setting confirmFlag of *S_k* as "true", adding it to *discoverList* and drawing it in the map. Jumping to Step8.

8.2 If there's no match between each MAC address in *A_{ij}* and that of every device in *switchList*, finding a matching device 'h' in *deviceList*. If succeeding, port *S_{ij}* is directly connected to 'h'. Drawing it in map and go to Step7.

Step9. end.

4. CONCLUSIONS

In an intelligent network management system we implement the discovery algorithm presented above, which is able to automatically inspect the subnet in a switch domain after finding out the backbone topology to discover link layer topology structure and display the corresponding topology

map. It is also able to start link layer topology discovering directly by restricting to the given switch domain and subnet. Complex relationships of ports and distributions of router-switch, switch-switch and switch-host can be captured exactly. Since the algorithm needs additional communication cost to ensure the integrality of MAC addresses switch ports learnt, it is better used for link layer topology discovery of mid-load subnets. Due to IP subnets widely run SNMP and spanning tree protocol, such algorithm is supposed to be the most popular, simplest and most effective means for data link layer topology discovery.

5. REFERENCES

- [1] K.McCloghrie and M.Rose. Management Information Base for Network Management of TCP/IP-based internets: MIB-1. Internet RFC-1213. Mar. 1991
- [2] A. Bierman, K.Jones. Physical Topology MIB RFC-2922. Sept. 2000
- [3] Paul Congdon. Link Layer Discovery Protocol and MIB. V1.0 May 20. 2002
- [4] Yigal Bejerano, Minos Garofalakis, etc. Topology Discovery in Heterogeneous IP Networks. IEEE INFOCOM 2000
- [5] Bruce Lowekamp, David R.O'Hallaron, etc. Topology Discovery for Large Ethernet Networks. SIGCOMM'01, August 27-31,2001,USA
- [6] Zheng hai and Zhang Guo-qing. An Algorithm for Physical Network Topology Discovery. Journal of computer research and development. March 2002. pp264-268.
- [7] William Stallings. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. Addison-Wesley Longman, Inc. 1999.(Third Edition)
- [8] E.Decker, P.Langille, etc. Definitions of Managed Objects for Bridges. Internet RFC-1493. July 1993

Liu Yuhua: associate professor, Department of Computer Science, Central China Normal University, Wuhan, China 430079 E-mail box: yhliu@ccnu.edu.cn

The Evaluation of the Probabilistic Packet Marking for Path Bifurcation *

Fu Jianming, Zhu Qin, Zhang Huanguo

School of Computer Science, the State Key Lab of Software Engineering of Wuhan University, 430072 Wuhan, China

Email: fujms@public.wh.hb.cn

ABSTRACT

At present the algorithm of probabilistic packet marking for IP trace back is used to handle the Denial of Service, and its assumption is that the attack traffic along an attack path is uniform. But this assumption is not always true because of routing asymmetry and exploiting source-routing, which results in the path bifurcations. In this paper, two cases of path bifurcations are investigated, and the relationship between the convergence capacity and the bifurcating probability of traffic is presented. Furthermore, the bifurcating probability of traffic is determined when the attack capacity is minimized. Especially when the traffic on some bifurcation path is sparse, the quantity of attack packets for the reconstructing the path will increase rapidly. Finally, our experimental results demonstrate the distribution of the bifurcating probability of traffic and the convergence capacity. Besides, its optimal bifurcating probability increases with the growth of the length of a bifurcating path, but its optimal marking probability decreases for this path. At the same time, these conclusions are consistent with theoretical value.

Keywords: IP traceback, Probabilistic packet marking, Path bifurcation, Denial of Service, Network security.

1. INTRODUCTION

Denial of Service (DOS) is the behavior of attackers to intentionally obstruct legitimate users from utilizing certain service, since DOS depletes a large quantity of server and the link resources. The difficulty in resolving DOS problem lies in that attackers always use counterfeit or illegal IP source addresses. Usually the Ingress filtering/Egress filtering techniques [2][3] can be used to filtrate attack packets, but the effect depends on the accuracy of recognizing attack packets and the extent to which the technique is applied. Presently, there are many researches on the algorithms of Probabilistic Packet Marking (Denoted PPM), including the marking probability of packets [4][5], the coding of marking information [4][6], the carrier of coding information [4][7][8], and the attack capacity of the reconstructed path [4][5]. In DOS and DDOS (Distributed DOS), it is commonly assumed that there is no bifurcation in an attack path, which means the attack traffics are equivalent for each router on this path. But in practical network environment, the traffics through each router on an attack path are more likely to be different because of the asymmetry of network routes [9] and the possible use of source-routing technique [10] by an attacker. It results in the increase of the attack capacity for the path reconstruction. In this paper, the reconstruction of an attack

path with bifurcations is mainly investigated, and then the experimental results are given.

2. DESCRIPTION OF THE PROBLEM

An attack network is depicted by the figure $G = (V, E)$, in which V is a set of network nodes and E is a set of edges between nodes. An attack paths $A = \{s, v_{12}, v_{21}, v_3, v_4, t\}$, where s is the node from which the attack originates, and t is a victim. If s is not spoofed, and we assume that routing is symmetric and fixed, then this attack path can be found out by means of trace route [11] or other techniques [12]. In practical attacks, s is counterfeit, and there exists two cases. IP source is invalid, so the IP host is unreachable probed by conventional technique. Another case is the IP source is reachable but illegal, which indicates that the attacker has embezzled other host's IP, and the result given by conventional probing technique is not A .

Usually, how to identify attack path or attacker's IP source is usually called IP tracing back problem. The PPM can effectively handle the problem of IP traceback [4]. When the attack packets pass a router, the router samples the packets according to a fixed probability p , and sampled packets and this router's IP address are coded into the marked packets. After the victim detects this invasion, it can reconstruct the attack path from those marked packets. In a general way, the convergence capacity is the minimum number of attack packets that are used to reconstruct the attack path at the victim.

3. BASIC PROBABILISTIC PACKET MARKING

Since the convergence capacity is related to Coupon Collector's Problem [13], we firstly introduce this problem, and then we investigate the PPM [4]. There is n kinds of the coupons, and the probability for each kind to appear is assumed to be equivalent, moreover N is the sampling times needed to acquire all n kinds of the coupons, so the minimum value of N is $n \ln(n)$.

Assume an attacker path is $A = \{s, R_d, R_{d-1}, \dots, R_2, R_1, t\}$. The marking probability of the router R_i is p_i . For the sake of generalization, let $p_i = p$, and p be a constant between 0 and 1, here $i \in [1, d]$. Because the routers near the victim may re-mark the packets previously marked by routers distant from the victim, the marking probability of each router at the victim is different. A_i indicates the marking probability of router R_i at t , and $A_i = p(1-p)^{i-1}$, $i \in [1, d]$.

According to the definition of A_i , we can deduce that $A_d < A_{d-1} < \dots < A_2 < A_1$. The marking probability of each router at t is estimated conservatively as A_d according to the coupon collector principle, and the convergence capacity N is

* This work is supported by Nation Science Foundation of China(90104005) and the project of HuBei Science Foundation(2002AB0037). Fu Jianming, Phd., associate professor, the main interests is network security; Zhu qin, graduate student, her interest is network security; Zhang Huanguo, professor, his interest is cryptology and information security.

$N(p, d) = \frac{\ln(d)}{p(1-p)^{d-1}}$. In order to acquire the extreme value of N as to p , let $\frac{\partial N}{\partial p} = 0$, the result is p equals to $1/d$, so the convergence capacity is $d \ln(d) e^{1-1/d}$.

4. PROBABILISTIC PACKET MARKING ON BIFURCATION PATHS

Consider the simple case of a path bifurcation, there are only two bifurcation paths l_1 and l_2 , and the ratio of their length l_1 and l_2 is $1/1$, $A = \{s, R_d, R_{d-1}, \dots, \{R_{kl}\}_{l1} | \{R_{k2}\}_{l2}, \dots, R_2, R_1, t\}$.

Let the probability of attack traffics passing R_{kl} be ρ_1 , and the probability of attack traffic passing R_{k2} be ρ_2 , so the probability of marking each router at t is as follows:

$$A_i = \begin{cases} p(1-p)^{i-1}, & i \neq k1, k2 \\ \rho_1 p(1-p)^{k-1}, & i = k1 \\ \rho_2 p(1-p)^{k-1}, & i = k2 \end{cases}$$

Here, $\rho_1 + \rho_2 = 1$.

The minimum value of A_i is

$$\alpha = \min\{A_i\} \approx \rho_1 \rho_2 p(1-p)^{d-k}(1-p)^{k-1},$$

Thus the convergence capacity N is

$$N(p, d) = \frac{\ln(d+1)}{\alpha} \approx \frac{\ln(d+1)}{\rho_1 \rho_2 p(1-p)^{d-1}}.$$

In order to get the minimal N for ρ_1 , let $\frac{\partial N}{\partial \rho_1} = 0$, so the

result is ρ_1 is equal to $1/2$, and N has the minimum value. In order to generalize the case, we consider r such simple bifurcation paths ($r \geq 2$). That is to say there are r paths from R_{k+1} to R_{k-1} . Let the probability of passing path i be ρ_i , and i is between 1 and r , so the convergence capacity is:

$$N(p, d) = \frac{\ln(d+r-1)}{\prod_{i=1}^r \rho_i p(1-p)^{d-1}} \quad (1)$$

ρ_i is $1/r$ while N is minimal according to Lagrange multiplier method, and $i \in [1, r]$. The more r is, the large N is. Hence, the number of bifurcation paths may sharply affect the convergence capacity.

To consider another path bifurcation, an attack path has also two bifurcation paths l_1 and l_2 , but the ratio of the length of l_1 and l_2 is $2/1$, and $Path = \{s, R_d, R_{d-1}, \dots, \{R_{kl1}, R_{kl2}\}_{l1} | \{R_{k2}\}_{l2}, \dots, R_2, R_1, t\}$.

Let the probability of attack traffics passing R_{kl} be ρ_1 , and the probability of attack traffic passing R_{k2} be ρ_2 , so the marking probability for each router at t is as follows.

$$A_i = \begin{cases} p(1-p)^{i-1}, & i \in [1, k-1] \\ \rho_1 p(1-p)^{k-1}, & i = k12 \\ \rho_1 p(1-p)^k, & i = k11 \\ \rho_2 p(1-p)^{k-1}, & i = k2 \\ p(1-\rho_1 p)(1-p)^{d-1}, & i \in [k+1, d] \end{cases}$$

Here, the sum of ρ_1 and ρ_2 is equal to 1.

The minimum value of A_i is

$$\alpha = \min\{A_i\} \approx \rho_1(1-\rho_1 p)\rho_2 p(1-p)^{d-1} \approx \rho_1 \rho_2 p(1-p)^d,$$

Thus the convergence capacity N is

$$N(p, d) = \frac{\ln(d+2)}{\alpha} \approx \frac{\ln(d+2)}{\rho_1 \rho_2 p(1-p)^d}.$$

In order to acquire the extreme value of N for ρ_1 , let $\frac{\partial N}{\partial \rho_1} = 0$,

so the result is ρ_1 is equal to $1/2$, and N has the minimum value. In order to generalize the case, let the length of the first bifurcation path be m , the probability of passing this path be ρ_1 , and the length of the second bifurcating path be 1, the probability of passing this path be ρ_2 , so the convergence capacity is:

$$N(p, d) \approx \frac{\ln(d+m)}{\rho_1 \rho_2 p(1-p)^{d+m-2}} \quad (2)$$

When m increases, N increases correspondently. When N gets its minimum value, the value of p is:

$$p = \frac{1}{d+m-1} \quad (3)$$

Additionally, in order to investigate the effect of ρ_1 on the extreme value of N , we only need to analyze the marking probability of the first node in a bifurcation path at the victim, because this marking probability is the minimum value of all probabilities along this bifurcation path.

$$l1: \rho_1 p(1-p)^{m-1} \cdot (1-p)^{k-1};$$

$$l2: \rho_2 p \cdot (1-p)^{k-1} \quad \text{and} \quad \rho_1 + \rho_2 = 1.$$

In order to make the above two nodes mentioned above be marking with the same probability, the two probabilities mentioned above must be equivalent.

$$\rho_1 p(1-p)^{m-1} \cdot (1-p)^{k-1} = \rho_2 p \cdot (1-p)^{k-1}$$

Therefore, the extreme value of ρ_1 is:

$$\rho_1 = \frac{1}{(1-p)^{m-1} + 1} \quad (4)$$

And when $m \geq 1$, N gets its minimum value.

In the formula (1) and (2), the worst situation is $\lim_{\rho_i \rightarrow 0} N = \infty$. When the probability of the traffic through a

bifurcation path is close to 0, the reconstruction of entire attack path is very difficult, because the convergence capacity approaches infinite.

5. EXPERIMENTAL ANALYSIS

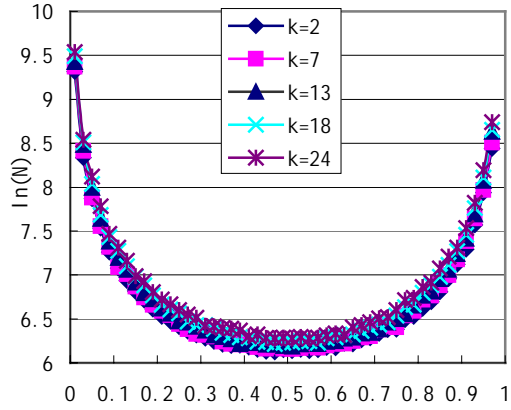
In order to investigate the marking problem of path bifurcations mentioned above, we design a simulation experimental environment to verify the correctness of our analysis. Assume that the marking probability of each router is equivalent to p , and each router marks all the passing packets according to edge-marking algorithm [4]. Fragment of marking information about a router is not considered for the simplicity.

Suppose there are two downstream routers R_{kl} and R_{k2} , the router R_i chooses the first one according to a certain probability ρ_1 . Hence the algorithm to choose a downstream router is described below.

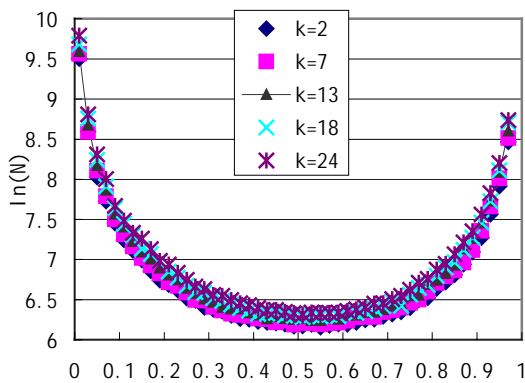
- (1) For each attack packet w , let x be a random number in domain $[0..1)$.

- (2) if $x < p_1$, then R_i transmits w to R_{kl} ; else R_i transmits w to R_{k2} .

Let the lengths of two bifurcation paths be l_1 and l_2 respectively, and the ratio of l_1 to l_2 be $m/1$, where m is a variable. The result depicted below is the average value of 1000 tests.



(a) $l_1 : l_2 = 1 : 1$



(b) $l_1 : l_2 = 2 : 1$

Fig.1. Relationship between convergence capacity and bifurcating probability

5.1 Convergence Capacity for Path Bifurcations

In our experiments, let $d=25$, $p=0.01$, $p_1=0.01\sim 0.97$, and $k=24, 18, 13, 7, 2$ respectively, where k is the distance between the originated node of the bifurcation path and the victim. The experimental results are illustrated in Fig.1 for the distinct ratio of l_1 to l_2 .

5.2 Effect of the Length of a Bifurcation Path on the Convergence Capacity

In order to study the relationship between the length of a bifurcation path and the convergence capacity, let m be 1, 2, 3, 4, 5, 6, 7, 8 respectively. The other parameters are assigned as $d=25$, $p=0.01$, $p_1=0.3$, $k=23$. The result is illustrated in the Fig.2, which accords with the theoretical value.

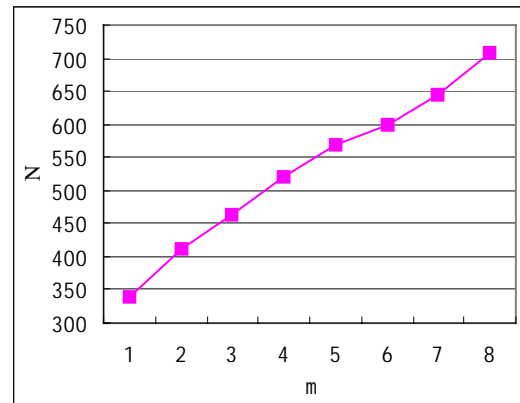


Fig.2. Relationship between m and N .

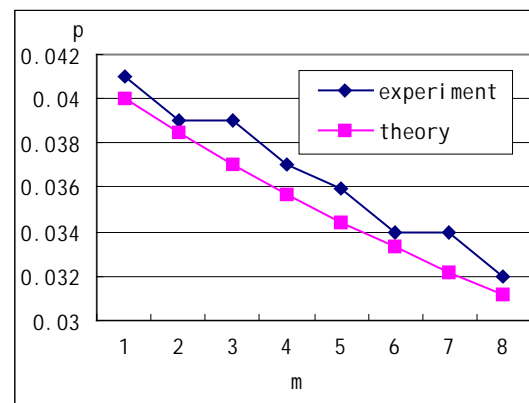


Fig.3. Relationship between m and p

5.3 Effect of the Length of a Bifurcation Path on the Optimal Marking Probability

Given the length of a bifurcation path m , the optimal marking probability p is a probability while the convergence capacity is minimal. That is to say, the convergence capacity increases when a marking probability is less or more than the p .

Let $d=25$, $p_1=0.3$, $k=23$, and Fig.3 describes the relationship between the optimal marking probability and the length of a bifurcation path for different m . And the experimental value is greater than the theoretical one. The reason is that at the victim node, each node is marked with different probability, whereas all these probabilities are equivalent for the formulae (3).

5.4 Effect of the Length of a Bifurcating Path on the Optimal Bifurcating Probability

Given the length of a bifurcation path m , the optimal bifurcating probability bp is a probability when the convergence capacity is minimal. In other words, the convergence capacity increases when a bifurcating probability is different from the optimal.

Let $d=25$, $p=0.04$, $k=23$, and fig.4 describes the relationship between the length of a bifurcation path and the optimal bifurcating probability. The optimal bifurcating probability increases with the growth of the length of the bifurcation path. And the experimental value is greater than the theoretical one, because the actually marked probability for each bifurcating

router at the victim is different, but the theory assumes it is identical.

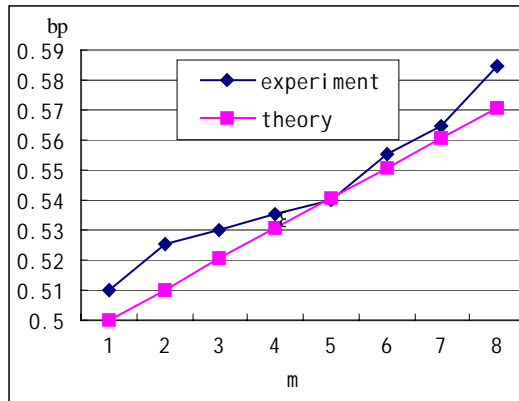


Fig.4 Relation between m and bp

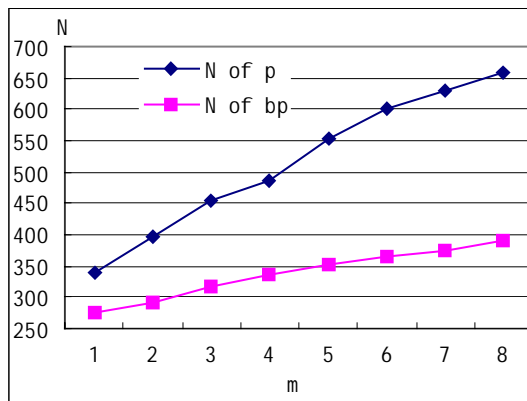


Fig.5 Relationship between m and N for p and bp

Fig.5 indicates that the minimum convergence capacity gradually increase with the increase of the length of the bifurcation path. That is to say, compared with the bifurcating probability, the marking probability has a greater effect on the convergence capacity, because the marking probability has effect on each node along the entire attack path, whereas the bifurcating probability affects the nodes along the bifurcation path only.

6. CONCLUSION

In this paper, two cases of path bifurcations are analyzed, and our conclusions are shown as follows. Firstly the relationship between the convergence capacity and the bifurcating probability is presented, and the more the number of bifurcation paths are, the greater the convergence capacity is. Furthermore, when the traffic traversing a bifurcation path is extremely uneven, it is difficult to reconstruct the complete attack path. Finally, the experimental results verify our conclusion, and its optimal bifurcating probability increases with the growth of a bifurcation path length, but its optimal marking probability decreases for this path.

In order to reconstruct the entire attack path in a short time, the attack path bifurcation should be taken into consideration. Our future work will be oriented to other path bifurcations in real attack paths.

7. REFERENCES

- [1] Denial of Services Attacks. CERT Coordination Center, Oct. 1997. Available at: http://www.cert.org/tech_tips/denial_of_service.html.
- [2] P.Ferguson and D.Senie. "Network Ingress Filtering:Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", RFC2827,2000.
- [3] T. Peng, C. Leckie and R. Kotagiri. "Protection from Distributed Denial of Service Attack Using History-based IP Filtering." In Proceedings of the IEEE International Conference on Communications (ICC 2003), 11-15 May 2003, Anchorage, Alaska, USA.
- [4] Stefan Savage,David Wetherall,Anna Karlin,and Tom Anderson, "Practical Network Support for IPTraceback", Proceedings of the 2000 ACM SIGCOMM Conference,August 2000:295-306.
- [5] T. Peng, C. Leckie and R. Kotagiri. "Adjusted Probabilistic Packet Marking for IP Traceback." In Proceedings of the Second IFIP Networking Conference (Networking 2002) 19-24 May 2002, Pisa, Italy.
- [6] Drew Dean, Matt Franklin, and Adam Stubblefield, "An algebraic approach to ip traceback," in *Network and Distributed System Security Symposium, NDSS '01*, February 2001.
- [7] Steven M.Bellovin,Editor, "ICMPTraceback Messages", Internet Draft: draft-ietf-itrace-03.txt , July.2003.
- [8] Vadim Kuznetsov, Helena Sandström, Andrei Simkin: An Evaluation of Different IP Traceback Approaches. ICICS 2002: 37-48. LNCS 2513.
- [9] Vern Paxson, "End-to-End Routing Behavior in the Internet," in *IEEE/ACM Transactions on Networking*, Oct 1997, vol. 5, pp. 601--615.
- [10] W. Richard Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994.
- [11] G. Malkin, "Traceroute Using an IP Option," RFC1393, January 1993.
- [12] Govindan,R.and Tangmunarunkit,H., "Heuristics for Internet Map Discovery," In *Proceedings of the 2000 IEEE INFOCOM Conference*, Tel Aviv, Israel, Mar. 2000.
- [13] W. Feller. *An Introduction to Probability Theory and Its Applications*, volume vol. I. Wiley, New York, 1967.

A Multicast Routing Optimization Algorithm with Bandwidth and Delay Constraints Based on GA*

Sun Baolin^{1,2} Li Layuan¹ Ma Jun²

¹. Department of Computer Science, Wuhan University of Technology
Wuhan 430063, P. R. China

². Department of Mathematics and Physics, Wuhan Institute of Science and Technology
Wuhan 430073, P. R. China

Email: sun0163@163.com Tel.: +86 (0)27- 62509828

jwtu@public.wh.hb.cn Tel.: +86 (0)27- 86534381 majun6367@hotmail.com Tel.: +86 (0)27- 87181075

ABSTRACT

With the rapid development of Internet, mobile networks and high-performance networking technology, QoS multicast routing in networks with uncertain parameters has become a very important research issue in the areas of networks and distributed systems. This is also a challenging and hard problem for the next generation Internet and high-performance networks. It attracts the interests of many people. This paper proposes a new multicast routing optimization algorithm based on Genetic Algorithms, which finds the low-cost multicasting tree with bandwidth and delay constraints. The simulations results show that the proposed algorithm is able to find a better solution, fast convergence speed and high reliability. It can meet the real-time requirement in multimedia communication networks. The scalability and the performance of the algorithm with increasing number of network nodes are also quite encouraging.

Keywords: QoS, multicast routing, genetic algorithm, routing optimization.

1. INTRODUCTION

Multicast services have been used by a variety of continuous media applications. For example, the multicast backbone (Mbone) of the Internet has been used to transport real time audio/video for news, entertainment, video conferencing, and distance learning [1-3,6-8,11,14]. The provision of Quality-of-Service (QoS) guarantees is of utmost importance for the development of the multicast services. Multicast routing has continued to be a very important research issue in the areas of networks and distributed systems. It attracts the interests of many people.

QoS multicast routing relies on state parameters specifying resource availability at network nodes or links, and uses them to find paths with enough free resources [1-3,12,14]. In turn, the successful routing of new flows together with the termination of existing ones, induce constant changes in the amount of resources available. These must then be communicated back to QoS multicast routing. Unfortunately, communicating such changes in a timely fashion is expensive and, at times, not even feasible [2,4,5,6]. As a result, changes in resources availability are usually communicated either infrequently or uncertainly. There are two main components to the cost of timely distribution of changes in network state: the

number of entities generating such updates and the frequency at which each entity generates updates.

In network operation, parameters about the state of nodes or links may be lost. This loss of certainty in state information can have a substantial impact on the multicast routing. For route selection, the main consequence of this loss of accuracy in network state information is that it now needs to consider not only the amount of resources that are available, but also the level of certainty with which these resources are indeed available. The second contributor to the cost of maintaining certain parameters information is the frequency of state changes, and therefore updates. Specifically, each advertisement of a state change consumes both network bandwidth on all the links over which it is sent, and processing cycles at all nodes where it is received. Keeping this overhead to a minimum is, therefore, desirable, if not mandatory. There are many different methods that can be used to achieve such a goal (see [2,3,5,6] for in-depth investigations of this issue and its impact on QoS routing), but they typically involve waiting for either a large enough change or a minimum amount of time passed.

Providing a comprehensive set of solutions for computing good paths in the presence of uncertainty is a daunting task that goes well beyond the scope of a single paper. This paper mainly present a multicast routing optimization algorithm with bandwidth and delay constraints based on Genetic Algorithm which can be suitable to the networks with uncertain parameters. The focus is on determining multicast routes from a source to a set of destinations with strict end-to-end delay requirements and minimum bandwidth available. Though the path determination problem with a single optimization parameter can be solved in *polynomial time*, the uncertainty of precise values of multiple objective functions make the problem a *NP-hard* [1-5]. The goal of this paper is to develop an algorithm to find out multicast routes with bandwidth and delay constraints by simultaneously optimizing end-to-end delay and bandwidth provisioning for guaranteed QoS.

The rest of the paper is organized as follows. Section 2 introduces a network model. Section 3 presents the multicast routing optimization algorithm based on Genetic Algorithms. Some simulation results are provided in Section 4. The paper concludes and future research in Section 5.

2. NETWORK MODEL

A network is usually represented as a weighted digraph $G = (N, E)$, where N denotes the set of nodes and E denotes the set of communication links connecting the nodes. $|N|$ and $|E|$ denote the number of nodes and links in the network respectively,

*The work is supported by National Natural Science Foundation of China (60172035, 90304018), NSF of Hubei Province (2000J154), Key Scientific Research Project of Hubei Education Department (2003A002) and NSF of Wuhan Institute of Science and Technology (20032418).

Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes [1-3,7,11].

We consider the multicast routing problem with bandwidth and delay constraints from one source node to multi-destination nodes.

Let $M = \{n_0, u_1, u_2, \dots, u_m\} \subseteq N$ be a set of form source to destination nodes of the multicast tree. Where n_0 is source node, $U = \{u_1, u_2, \dots, u_m\}$ be a set of destination nodes. Multicast tree $T = (N_T, E_T)$, where $N_T \subseteq N$, $E_T \subseteq E$, there exists the path $P_T(n_0, d)$ from source node n_0 to each destination node $d \in U$ in T [1-3,11].

Definition 1: The cost of multicast tree T is $C(T) = \sum_{e \in E_T} C(e)$.

Definition 2: The bandwidth of multicast tree T is the minimum value of link bandwidth in the path from source node n_0 to each destination node $d \in U$. i.e.

$$B(T) = \min(B(e), e \in E_T).$$

Definition 3: The delay of multicast tree T is the maximum value of delay in the path from source node n_0 to each destination node $d \in U$. i.e.

$$D(T) = \max(\sum_{e \in P_T(n_0, d)} D(e), d \in U).$$

Definition 4: Assume the minimum bandwidth constraint of multicast tree is B , the maximum delay constraint is D , given a multicast demand R , then, the problem of bandwidth-delay constrained multicast routing is to find a multicast tree T , satisfying:

- (1) Bandwidth constraint: $B(T) \geq B$.
- (2) Delay constraint: $D(T) \leq D$.

Suppose $S(R)$ is the set, $S(R)$ satisfies the conditions above, then, the multicast tree T which we find is:

$$C(T) = \min(C(T_s), T_s \in S(R))$$

3. GENETIC ALGORITHMS

Genetic algorithms are based on the mechanics of natural evolution. Throughout their artificial evolution, successive generations each consisting of a population of possible solutions, called individuals (chromosomes), search for beneficial adaptations to solve the given problem. This search is carried out by employing the Darwinian principles of "reproduction and survival of the fittest" and the genetic operators of crossover and mutation which derive the new offspring population from the current population. Reproduction involves selecting, in proportion to its fitness level, an individual from the current population and allowing it to survive by copying it to the new population of individuals. The individual's fitness level is usually based on the cost function given by the problem (e.g., QoS multicast routing) under consideration. Then, crossover and mutation are carried on two randomly chosen individuals of the current population creating two new offspring individuals. Crossover involves swapping two randomly located sub-chromosomes of the two mating chromosomes. Mutation is applied to randomly selected genes, where the values associated with such a gene is

randomly changed to another value within an allowed range. The offspring population replaces the parent population, and the process is repeated for many generations. Typically, the best individual that appeared in any generation of the run is designated as the result produced by the genetic algorithm.

Encoding Representation

In genetic algorithms, the critical problem is how to transform the solution of the problems to the chromosomes which represents with encoding. The chromosomes of genetic algorithms is composed of a series of integral queuing and the encoding method based on routing representation, which the most natural and simplest representing method. Given a source node n_0 and destination nodes set $U = \{u_1, u_2, \dots, u_m\}$, a chromosome can be represented by a string of integers with length m . The chromosome of genetic algorithms is composed of a series of integral queuing with length m , the gene of genetic algorithms is the path in path set $\{P_i^1, \dots, P_i^j, \dots, P_i^l\}$ [11,12] between n_0 and u_i , where, P_i^j is the j -th path of destination node u_i , l denotes the path number between n_0 and u_i . Each chromosome in population denotes a multicast tree. This coding method was first proposed in reference [12] for the point-to-point routing problem. Obviously, a chromosome represents a candidate solution for the multicast routing problem since it guarantees a path between the source node and any of the destination nodes. The major advantage of using the coding method of reference [12] is that given a chromosome, the links of the multicast tree can be easily identified and the path delay or bandwidth can be taken into consideration through the proper selection of routes in routing tables. Since there are so many paths between node n_0 and u_i , such that the encoding space of chromosomes possibly becomes larger, which decreases the convergence of solution. Now for each destination node $d \in U$, by the k -th the shortest route algorithm, the encoding space can be improved by finding out all routes that satisfy bandwidth constraint from source node n_0 to destination node $d \in U$ and composing routes set as candidate routes set of genetic algorithm encoding space. Assume that U_i is the set of destination node u_i which satisfies bandwidth constrained, then,

$$U_i = \{P_i^1, \dots, P_i^j, \dots, P_i^k\}, \quad k \leq l$$

Where, P_i^j denotes the j -th route which satisfies bandwidth constraint of destination node u_i . Choose arbitrarily a route from each route set U_i respectively, and compose the initial population of chromosomes. Obviously, the multicast tree covered all destination nodes, diminished bandwidth constraint in the algorithm and optimized the performance of networks, decreased searching space of the algorithm, diminished the probability which dissatisfied, bandwidth constraint link in algorithm selection, but satisfied the demand of bandwidth constraint.

Therefore, the chromosome of genetic algorithm can be made of a series of integral queuing, namely, the encoding method based on routing representation; this method decreased encoding space, also omitted decoding operation. The relationship among the chromosome, gene, and routing table is explained in Figure 1.

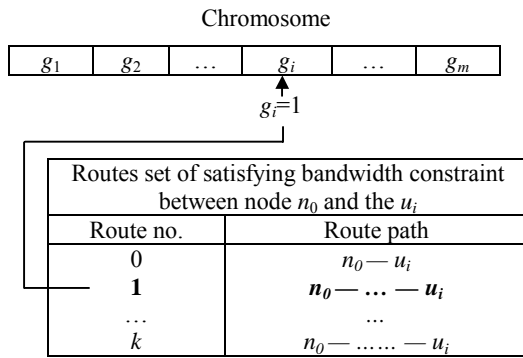


Figure 1. Representation of chromosomes

Fitness Sharing Function

The fitness function interprets the chromosome in terms of physical representation and evaluates its fitness based on traits of being desired in the solution. However, the fitness function must measure accurately the quality of the chromosomes in the population. The definition of the fitness function, therefore, is very critical.

Genetic algorithm uses rarely the outer information in evolution searching, instead it uses fitness function as basis and the values of every individual in population for searching. As a result, fitness function affects directly the convergent speed of genetic algorithm and whether the global optimum is found. The fitness value of a chromosome is the value of the fitness function for the solution (e.g., a multicast tree) represented by the chromosome. Given a initial population $H=\{h_1, h_2, \dots, h_p\}$, the fitness value of each chromosome is computed as follows. Let $C(h_i)$ be the sum of the costs of the links of the graph represented by the chromosome h_i and $C(L)$ be the sum of the costs of all the links in the network. The fitness value of the chromosome h_i , $F(h_i)$, is given by

$$F(h_i) = 1 - C(h_i)/C(L) \quad (0 \leq F(h_i) < 1)$$

Selection Operations

Selection operation is used to certain or crossover individuals and selected individual can produce many sub-individuals. Selection operation has two procedures: firstly, computing fitness value; secondly, queuing it from the biggest to the smallest, namely, $F(h_1) \geq F(h_2) \geq \dots \geq F(h_p)$, then, the max fitness value is the best individual, selecting the best individual as father-individual, the selection probability of each individual is proportional to its fitness value, the selected probability is higher when the individual fitness value is bigger. If the same chromosomes have been got, only one chromosome exists. The rest chromosomes can be canceled.

Crossover Operations

As the algorithm executes, at every iteration we get a set of *non-dominated* strings whose fitness values represent the *Pareto-optimal* solutions for that iteration. The crossover and mutation operations are the same as normal genetic algorithms. Anyhow these operations must not to produce any illegal paths. A close look into the structure of the chromosome in figure 1 reveals that these genetic operations cannot be performed on any arbitrary gene (network nodes), as that can give birth to some paths which do not exist at all.

Crossover examines the current solutions in order to find better ones [7-15]. Physically, crossover in the shortest path

routing problem plays the role of exchanging each partial-route of two chosen chromosomes in such a manner that the offspring produced by the crossover represents only one route. This dictates selection of one-point crossover as a good candidate scheme for the proposed GA. One partial-route connects the source node to an intermediate node, and the other partial-route connects the intermediate node to the destination node. The crossover between two dominant parents chosen by the selection gives higher probability of producing offspring having dominant traits.

But the mechanism of the crossover is not the same as that of the conventional one-point crossover. In the proposed scheme, two chromosomes chosen for crossover should have at least one common gene (node), but there is no requirement that they be located at the same locus. That is to say, the crossover does not depend on the position of nodes in routing paths. Figure 2 shows an example of the crossover procedure. As shown in figure 2, a set of pairs of nodes which are commonly included in the two (chosen) chromosomes without positional consistency are formed (i.e., (3,2) and (5,4)). Such pairs are also called "potential crossing sites". Then, one pair (i.e., (3,2)) is randomly chosen and the locus of each node becomes a crossing site of each chromosome. The crossing points of two chromosomes may be different from each other.

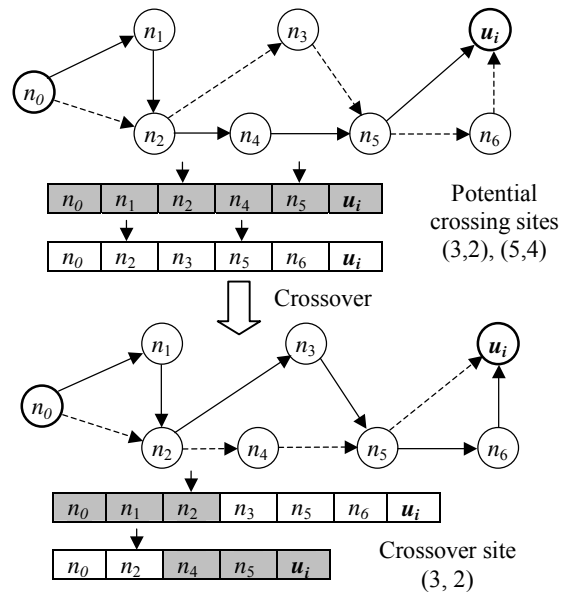


Figure 2. Overall procedure of the crossover

Mutation Operations

The population undergoes mutation by an actual change or flipping of one of the genes of the candidate chromosomes, which keeping away from local optima [7-15]. Physically, it generates an alternative partial-route from the mutation node to the destination node in the proposed GA. Topological information database is utilized for the purpose. Of course, mutation may induce a subtle bias for reasons indicated earlier.

Figure 3 shows the overall procedure of the mutation operation. As can be seen from Figure 3, in order to perform a mutation, a gene (i.e., node n_2) is randomly selected from the chosen chromosome (mutation point). One of the nodes, connected directly to the mutation point, is chosen randomly

as the first node of the alternative partial-route.

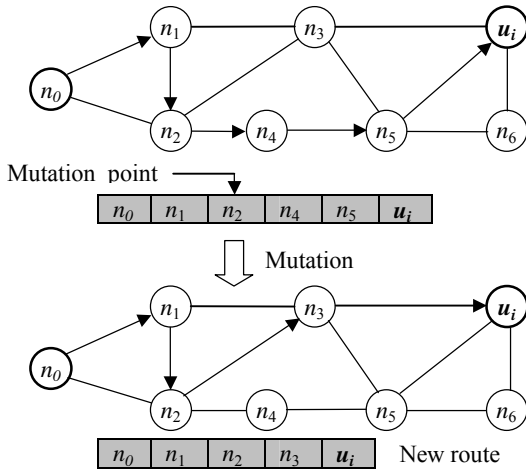


Figure 3. Overall procedure of the mutation

However, nodes already included in an upper partial-route should be deleted from the database so as not to include the same node twice in the new routing path. The upper partial-route represents the surviving portion of the previous route after mutation; which is the partial chromosome stretching from the first gene to the intermediate gene at the mutation point.

4. ALGORITHM ANALYSIS AND SIMULATIONS RESULTS

Analysis of Convergence

Theorem 5: The genetic algorithm proposed in this paper converges to the global optimal solution.

Proof: The genetic algorithm has following merits: (1) The method which uses the candidate routing set from source node to each destination node, makes the searching speed faster, and the whole process could be done in shorter time; (2) selecting by proportion and pertaining the optimal individual before selection; (3) Changeable length chromosome encoding method which based on routing expression is used; (4) Crossover probability between $[0,1]$; (5) Mutation probability between $[0,1]$, by theorem 2.7 in reference [15]: have crossover probability between $[0,1]$, mutation probability between $(0,1)$, at the same time, the genetic algorithm obtained according to the method proposed above can converge to the global optimal solution. Hence, the genetic algorithm proposed in this paper can converge to the global optimal solution.

Computing Complexity

The genetic algorithm is parallel algorithm, the searching speed is which fast, that the whole process can be done in short time. Thus, the time complexity of algorithm in this paper is decided by the time complexity of reference [4], which is $O(|E|+|N|\log|N|+k)$.

Simulations Results

Adopting the network topology structure as in figure 4 in simulation experiment, the characteristic of links or sides in figure can be represented by a triple-group (B_{ij}, D_{ij}, C_{ij}) , its

value randomly given. Assuming the source node n_0 is node 1, destination node set $U=\{4, 5, 7, 8\}$, the smallest bandwidth constraint $B=10$, by the algorithm for finding the k -th shortest paths in reference [4], we can find the candidate route set from source node 1 to each destination node, as in table 1.

Table 1. The candidate path set from source node to each destination node

Destination node	Candidate path set
4	$\{1,2,4\}, \{1,3,4\}, \{1,5,6,4\}$
5	$\{1,5\}$
7	$\{1,3,4,6,7\}, \{1,5,6,7\}, \{1,2,8,7\}, \{1,2,4,6,7\}$
8	$\{1,5,6,7,8\}, \{1,3,4,6,7,8\}, \{1,2,4,6,7,8\}, \{1,2,8\}$

In this genetic algorithm, let crossover probability be 0.9, mutation probability be 0.2. When bandwidth constraint $B=10$, delay constraint $D=7$, generated multicast tree as in figure 5. When bandwidth constraint $B=12$, delay constraint $D=8$, generated multicast tree as in figure 6.

We compare this algorithm with the one in reference [11]. Figure 7 compares desired route failure Probability vs. route failure ratio. Figure 8 denotes the convergence comparison of cost with operation algebra of multicast tree generated by the two algorithms, this algorithm can speedily generate the optimal solution, furthermore, its advantage is more obvious when network scale is bigger, and bandwidth constraint is amplified. Repeat the simulation with increasing number of network nodes and the efficiency of our algorithm. As the network becomes highly condensed, our algorithm exhibits a more linear and stable pattern than existing scalar optimization algorithm. This *approximate linearity* of the curve in figure 9 corroborates the scalability of the algorithm.

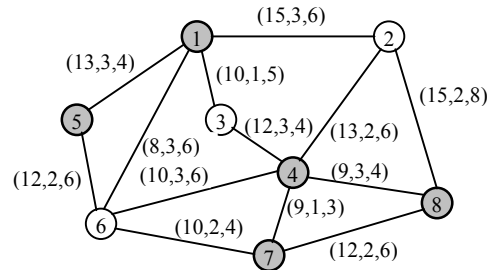


Figure 4. Network topology structure

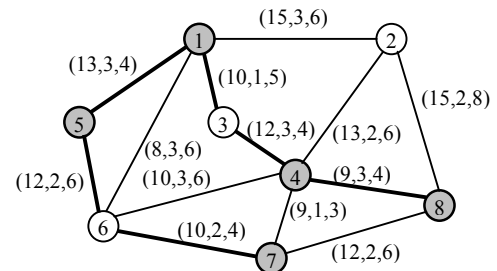


Figure 5. Genetic algorithm generate multicast tree ($B=10, D=7$)

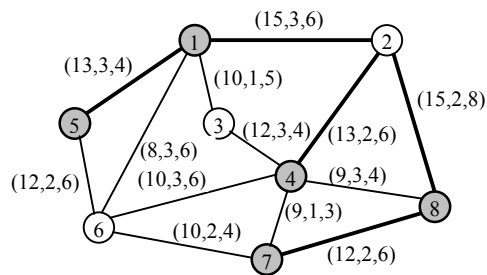


Figure 6. Genetic algorithm generate multicast tree ($B=12, D=8$)

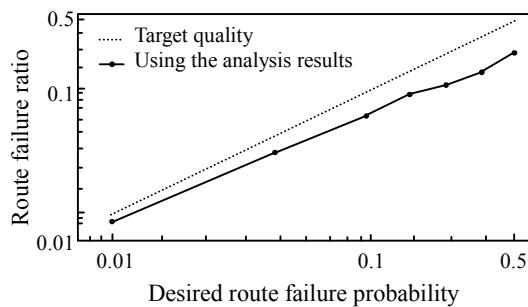


Figure 7. Desired route failure probability vs. Route failure ratio

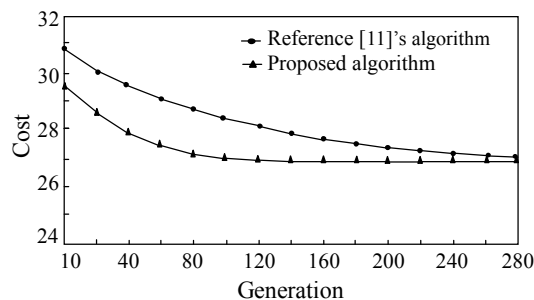


Figure 8. The effect of the cost with genetic generation

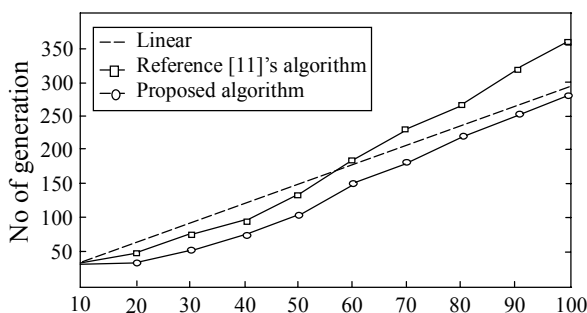


Figure 9. Performance of the algorithm with increasing number of nodes

5. CONCLUSIONS AND FUTURE WORK

This paper proposes a QoS multicast routing model and optimization algorithm based on bandwidth and delay constraints and gives the heuristic genetic algorithm of

minimum-costs QoS multicast tree and bandwidth-delay constraint. This algorithm has merits as below: (1) using the algorithm to find the k -th shortest path in reference [4] to pretreat, constructing candidate route set, decreasing searching space of algorithm efficiently, and increasing searching efficiency of algorithm marvelously; (2) simplified encoding operation based on the tree structure encoding of route, omitting complicated encoding-decoding process; (3) guaranteed and speeded searching ability of the optimal solution and the global convergence of solution by heuristic crossover and mutation operation; (4) the time complexity of the algorithm is $O(|E|+|N|\log|N|+k)$. Experiment represents: its convergent speed is fast and reliability high. Especially in big network, the algorithm can decrease greatly routing computation time, satisfying the topology structure of real time communication environment, high dynamic and the requirement of network structure of QoS multicast routing.

QoS constraint multicast routing is the foreland research project in networks and information technology field. Although many lovers have got better results in single constraint (especially delay constraint) multicast routing, the results based on multi-QoS constraint multicast routing are not so satisfying. This algorithm can expand to multi-QoS constraint's multicast routing problem based on this algorithm; the chromosome only of fitness function can be changed and the delay constraint can be improved so as to make the algorithm widely applied.

In a word, the deep research of QoS constraint multicast routing will increase the technology of high performance network routing system, and will be widely applied in video, multimedia broadcasting and distance education fields, etc.

6. REFERENCES

- [1]. Li Layuan and Li Chunlin, "The QoS routing algorithm for ATM networks", Computer Communications, Vol. 24, No.3-4, 2001, pp. 416-421.
- [2]. Li Layuan and Li Chunlin, "The QoS-based routing algorithms for high-speed networks", Proc of WCC, Aug. 2000, pp.1623-1628.
- [3]. Li Layuan and Li Chunlin, "A multicast routing protocol with multiple QoS constraints", Proc of WCC, Aug. 2002.
- [4]. D. Eppstein, "Finding the k shortest paths", SIAM J. Computing, Vol. 28, No. 2, 1998, pp. 652-673.
- [5]. F. Bauer and A. Varma, "Distributed algorithms for multicast path setup in data networks", IEEE/ACM Transaction on Networking, Vol.1, No.3, 1993, pp. 287-292.
- [6]. J. J. Wu and R.-H. Hwang, "Multicast routing with multiple constraints", Information Sciences, Vol. 124, 2000, pp. 29-57.
- [7]. Z. Xiawei, C. Changjia and Z. Gang, "A Genetic Algorithm for Multicasting Routing Problem", International Conference Communication Technology Proceedings, WCC-ICCT 2000, 2000, pp. 1248-1253.
- [8]. Zhang Q, Lenug Y. W, "An orthogonal genetic algorithm for multimedia multicast routing", IEEE Trans Evolutionary Computation, Vol. 3, 1999, pp. 53-62.
- [9]. M. Munemoto, Y. Takai, and Y. Sato, "A Migration Scheme for the Genetic Adaptive Routing Algorithm", IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp. 2774-2779.
- [10]. J. Inagaki, M. Haseyama, and H. Kitajima, "A Genetic

Algorithm for Determining Multiple Routes and Its Applications”, Proceedings of IEEE International Symposium on Circuits and Systems, 1999, pp. 137~140.

- [11]. R. -H. Hwang, W. -Y. Do and S. -C. Yang, “Multicast Routing Based on Genetic Algorithms”, Journal of Information Science and Engineering, Vol. 16, 2000, pp. 885~901.
- [12]. N. Shimamoto, A. Hiramatu and K. Yamasaki, “A dynamic routing control based on a genetic algorithm”, 1993 IEEE International Conference on Neural Network, 1993, pp. 1123~1128.
- [13]. F. Xiang, L. Junzhou. W. Jieyi and G. Guanqun, “QoS routing based on genetic algorithm”, Computer Communications, Vol. 22, 1999, pp. 1392~1399.
- [14]. Wang Xinhong and Wang Guangxing, “A Multicast Routing Approach with Delay-Constrained Minimum-Cost Based on Genetic Algorithm”, Journal of China Institute of Communications, Vol. 23, No. 3, 2002, pp. 112~117.
- [15]. Chen Guoliang et al, Genetic Algorithm and Its Application. Beijing: People’s Posts and Telecommunications Press, 1996(in Chinese).

A Kind of Low Latency Communication Way over Ethernet

DAI Xinfu^{1,2}, OU Zhonghong¹, FANG Ming¹, YUAN Youguang²

¹College of Computer Science and Technology, Harbin Engineering University, Harbin, China 150001

²Wuhan Digital Engineering Institute, Wuhan, China 430074

Email: daixinfu@sina.com

Tel: +86 (0)27 87534269

ABSTRACT:

In PC cluster systems, some kinds of multiprocessor systems and distributed systems, the low latency property of communication is a research focus. Ethernet applied to these systems should be very satisfying with cost and performance, if its low latency property could be ensured by a kind of communication way. The paper presents a kind of Low Latency Communication way over Ethernet (LLCE) that reduces the overhead and complicity of communication software. The solution is to bypass TCP/IP protocol within kernel context and directly program the Ethernet interface controller. LLCE has achieved lower latency and higher bandwidth than TCP/IP communication over same 1000Mb/s Intel PRO/1000 XF Ethernet adapter on PCs with 2.0GHz Pentium CPU and 133MB/s PCI.

Key words: Ethernet, latency, TCP/IP protocol, software overhead, bypass.

1. INTRODUCTION

It is absolutely necessary for the interior network to possess the low latency property with high bandwidth in PC cluster systems, some kinds of multiprocessor systems and distributed systems. These systems usually adopt fast-speed networks such as Ethernet [1], ATM [2], Myrinet [3] with TCP/IP [4] communication protocol, in order to achieve high performance communication. However, high bandwidth networks probably result in high latency, while low bandwidth networks have low latency sometimes. For example, average TCP/IP round trip time of 10M Ethernet is 1438 μ s, while 640M Myrinet's is 1506 μ s on Sun SPARC20 workstations [5]. In addition, average TCP/IP round trip time of 1000M Intel PRO/1000 XF is 220 μ s/512B, while 100M Realtek RTL8139D's is 174 μ s/512B on PCs with 2.0GHz Pentium CPU and 133MB/s PCI. Accordingly, the communication latency besides the hardware link bandwidth heavily constricts network performance.

Generally, network communication latency consists of protocol processing time on sender and receiver, data copying time, operating system switching time, data transmitting time, buffer and header managing time. Those fast-speed networks cannot still reduce the communication software overhead, even though they may shorten data transmitting time. Due to the heavy communication software overhead, the communication latency has become of the bottleneck of high performance network.

References [1]~[5] have described some low latency high performance communication environments over some types of network. It ought to be attractive for Ethernet to be applied to those systems in terms of cost and performance.

Aiming at this object and exploring the low latency communication protocol, the paper devises a kind of low latency communication way over Ethernet (LLCE).

The rest of the paper is organized as follows. Section 2 presents a way to improve network performance after categorizing the traditional TCP/IP communication latency, and provides a solution LLCE. Section 3 is the description of principle of LLCE. Section 4 analyzes the performance of LLCE. Section 5 draws a conclusion on the paper, and sets future work on LLCE.

2. TRADITIONAL TCP/IP COMMUNICATION LATENCY

TCP/IP protocol is a multi-level communication protocols stack. Different level protocols accomplish different communication functions. Their software processing overheads mostly come from follows [6]~[9].

Firstly, the traditional TCP/IP protocol holds complex functional modules. Belonging to Internet protocol, TCP/IP protocol offers a few complex multifunctional modules, such as flow control module, error control module, retransmission mechanism module, congestion control module and routing mechanism module. These complicated functional modules cannot but result in heavy overhead. Moreover, there are redundant functions among them, and the redundancies can be omitted under some applications.

Secondly, there are overheads when messages are copied between user context and kernel context on sending software and receiving software with TCP/IP protocol. Especially, the overheads become increasingly heavy as the size of message packet increases.

Thirdly, operating system switching accumulates delays. The system calls and primitives offered by operating system build up the basis of network protocol that involves context switching, page swap, I/O device initiation and interrupt response. The operating system overheads cannot be ignored.

Fourthly, the size of message packet is constrained by MTU (maximum transmit unit) on the TCP/IP communication, which causes IP protocol service to part big size message packet into several small size message packets on sending agent. And small size message packets usually reach receiver out of order along different route. Subsequently, the receiver need retrieve out-of-order small size message packets into the primal message packet. Parting and retrieving is always time-consuming for big size message packet.

Consequently, so far as latency performance is concerned,

TCP/IP protocol is unqualified for interior network of PC cluster, some kinds of multiprocessor and distributed system where communication nodes keep close, and the network is also reliable, and the small size messages always get to receiver orderly. In other ways, Ethernet is low-cost, and applying Ethernet to these systems should be very satisfying in terms of cost and performance. In order to realize this object, the paper puts forward a kind of low latency communication way over Ethernet (LLCE). The solution has applications bypass TCP/IP protocol within the kernel context, and directly program the Ethernet adapter. To decrease the Ethernet communication latency and increase the communication bandwidth, LLCE mainly reduces the overhead and complicity of communication software.

3. THE PRINCIPLE OF LLCE

In traditional TCP/IP communication, communication is accomplished one after the other by application program within user context invoking the system call, subsequently entering TCP/IP protocol stack, finally by the device driver sending message to network. The receiving procedure is done inversely (see figure 1). Communication overheads contain protocol software overheads and network device driver overheads on sender and receiver.

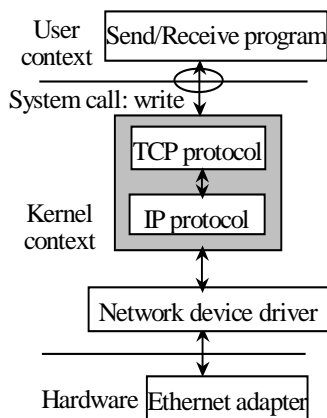


Figure 1 TCP/IP communication

To operate Ethernet adapter directly and program network adapter's registers, LLCE is mainly implemented through bypassing the influence of operating system on protocol stack. It makes Ethernet communication software bypass the complexly time-consuming TCP/IP protocol stack. Moreover, communication driver is also optimized. LLCE maps Ethernet adapter into char communication device under operating system management (see figure 2), while Ethernet adapter is defined into network communication device in the traditional TCP/IP protocol communication. Thus, LLCE can reduce the time overhead and complicity of communication software, and decrease the network communication latency. LLCE involves a few key technique points such as composing fragment, user communication interface.

3.1 Composing fragment

Fragment is the carrier of message that is processed in host and carried along network. The format of fragment varies with protocol level that the message reaches. TCP/IP

fragment carries data between different protocol services when messages are processed in host, while Ethernet MAC

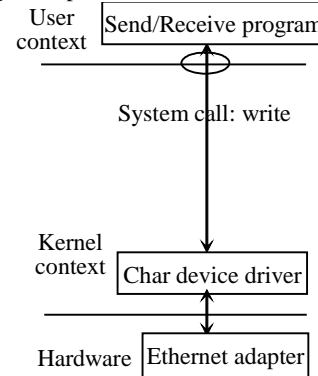


Figure 2 LLCE communication

fragment accomplishes to transmit data along network. Composing fragment means transforming upper-layer protocol fragment into under-layer protocol fragment in sending procedure. It is converse in receiving procedure. So the overheads of composing fragment consist of protocol processing, data copying, and buffer managing. Therefore, decreasing delays may be achieved by cutting down a few services procedure of the communication protocol correctly.

Table 1 TCP/IP+IEEE802.3 fragment

Destined MAC addr.	Source MAC addr.	Length	Data field		Fragment check
			IP header	Data	
6 Bytes	6Bytes	2Bytes	24Bytes	TCP 24Bytes User Data <1452Bytes	4 Bytes

Table 2 LLCE fragment

Destined MAC addr.	Source MAC addr.	Length	Data field			Fragment check
			D. port	S. port	User Data	
6 Bytes	6 Bytes	2Bytes	1Bytes	1Bytes	<1498Bytes	4 Bytes

The traditional TCP/IP protocol is a complicated multi-level protocol stack that contains a few complex redundant function modules. If user application needs to send data, firstly TCP protocol service composes user data unit into TCP protocol packet through appending TCP header to data unit; secondly IP protocol service composes TCP protocol packet into IP protocol packet through appending IP header to TCP protocol packet; thirdly, ARP protocol service must compose IP protocol packet through appending destined physical (Ethernet MAC) address corresponded to the destined IP address to IP protocol packet, and finishes user data unit into TCP/IP+IEEE802.3 fragment (See table 1); finally Ethernet network driver accomplishes to carry TCP/IP+IEEE802.3 fragment along network. The receiving procedure is converse. Therefore, assembling or disassembling TCP/IP+IEEE802.3 fragment is complicatedly time-consuming procedure, and to bypass TCP/IP protocol stack is certainly able to remarkably reduce the communication latency.

Owing to bypassing the TCP/IP protocol service, composing fragment in LLCE means simply packing user application data with Ethernet MAC protocol, which make message fit for maximum size or minimum size of MAC protocol fragment. Fragment format of LLCE shows in table 2.

Compared with TCP/IP+IEEE802.3 fragment, LLCE fragment has merely different data field. Its data field only contains user data, and isn't packed with upper-layer protocol headers. Hence, LLCE reduces the overheads that upper-layer protocols are assembled or disassembled, decreases the delay of header management and protocol service processing, and decreases the delay of copying data between user context and kernel context. Moreover, LLCE also increases the amount of user data carried along network, as improves the network communication actual bandwidth, and reduces latency of the big message.

3.2 User interface

In user context, LLCE communication processes are identified from port ID, and message packet consists of destined Ethernet MAC, source Ethernet MAC, destined process port, source process port and user data one after the other. Each communication process contains a message queue and a finish event queue about sending or receiving. On communicating, sending process firstly composes message packet, subsequently inserts message packet into message queue. Before beginning to communicate, receiving process must supply receiving message queue. After having received message, receiver brings about a message finish event to enter message finish event queue to notice receiving process. Analogically, after sending message, sender brings about a message finish event to enter message finish event queue to notice sending process.

In LLCE, application program uses file I/O to call special network communication interface that is supplied with the device file object that the special char device driver within operating system appoints for a file name in device file directory, for instance: /dev/EthernetAdapter. It corresponds to hardware Ethernet adapter. Operating Ethernet adapter only require using file I/O of operating system such as *write*, *read*, *open*, *close*, *ioctl*.

Sending routine:

```
Eth_fd=open ("/dev/EthernetAdapter", OPEN_MODE);
/*firstly initialize network card, begin to communicate*/
num_send = write (Eth_fd, send_buffer, length ); /*sending
message by network card*/
close (Eth_fd);
/*close network card, and end communicating*/
```

Receiving routine:

```
Eth_fd = open("/dev/EthernetAdapter", OPEN_MODE); /*
firstly initialize network card, begin to communicate */
num_receive = read( Eth_fd,recv_buffer, MAXDATASIZE );
/*receiving message by network card
*/
close (Eth_fd);
/*close network card, and end communicating*/
```

4. ANALYZING THE PERFORMANCE OF LLCE

The paper analyzes LLCE performance of 1000Mb/s Intel PRO/1000 XF Ethernet adapter on PCs with 2.0GHz Pentium CPU and 133MB/s PCI under Red Linux7.2 operating system. By measuring communication latency and bandwidth performance, the paper compares LLCE performance with TCP/IP communication performance. In addition, the paper achieves TCP/IP communication latency and bandwidth of 100Mb/s Realtek RTL8139D Ethernet card on this computing environment.

4.1 Measuring round trip time

The paper measures LLCE round trip time with ping-pong method. Sender sends a packet, and simultaneously starts to time. After receiving the packet, receiver promptly makes the packet rebound. Sender stops timing as soon as it receives the packet, and calculates the time difference. The procedure is repeated 100 times. Sender achieves round trip time through averaging the difference sum. The result is shown in figure 3.

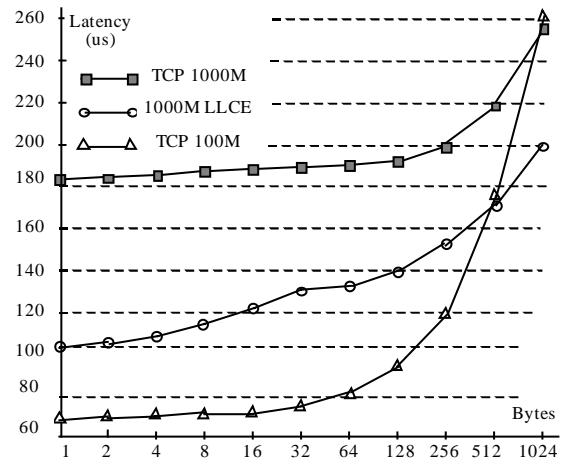


Figure 3 comparing round trip time

Communication buffer is 2048 bytes. TCP_NODELAY is set on measuring TCP/IP performance, which may reduce small size message packet latency.

1000M LLCE achieves 97μs round trip time on single byte message packet, while the round trip time of TCP 1000M is 182μs. It is still more than that of TCP 100M. For 512 bytes message latency, 1000M LLCE 163μs latency is less than TCP 100M 174μs latency. When it is 1024 bytes message, TCP 100M latency is the maximum in 260μs, while 1000M LLCE achieves the minimum 199μs latency and TCP 1000M gains 255μs latency.

The results primarily attribute to the fact that 1000M Intel PRO/1000 XF is more complex as compared with RTL 8139D. There are an amount of complex registers in Intel 82544 controller, which is the basis of 1000M Intel PRO/1000 XF as opposed to the 8139D controller. Therefore, the communication driver of 1000M Intel PRO/1000 XF spends much more time than 100M RTL8139D does. Especially, under same buffer management the overhead of communication driver program is the main source of communication latency for small message packet. It just about results in that latency of 1000M is more than 100M's for small size message communication. Compared with the traditional TCP/IP communication, 1000M LLCE only shortens the time of communication protocol processing, and partly reduces the communication latency of 1000M Intel PRO/1000 XF.

For big message communication, data transmitting and data copying between the buffer of network adapter and the memory of host (through DMA) are the primary source of communication latency. The two completely rely on network adapter hardware performance. 1000M Intel PRO/1000 XF

certainly excels 100M RTL8139D in hardware performance. Hence, the latency of 1000M is lower than 100M's, and 1000M LLCE's is the lowest on big message communication. Moreover, compared with small message packet, big message packet communication achieves more decrease in latency. This just proves that LLCE is efficient for improving on latency.

4.2 Measuring communication bandwidth

Communication bandwidth represents the capability of transmission data. The paper applies following method to measure bandwidth. Sender continuously sends 100 packets to receiver, and receiver starts to time as soon as it receives the first packet. After receiving all packets, receiver calculates communication bandwidth. Figure 4 shows the result.

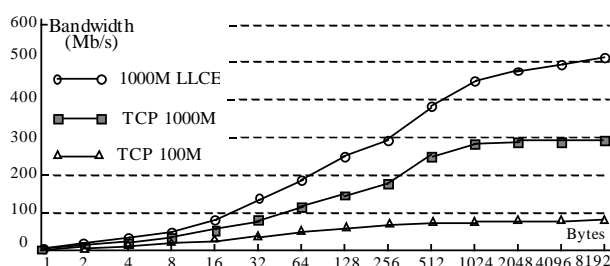


Figure 4 comparing bandwidth

1000M TCP Intel PRO/1000 XF Ethernet adapter almost reaches the limit of actual bandwidth in 263Mb/s on 1024 bytes message packet, while 100M TCP RTL8139D reaches the limit of actual bandwidth in 82Mb/s on 512 bytes message packet. 1000M LLCE markedly improves on the bandwidth, and reach 503Mb/s bandwidth on 8192 bytes message packet communication.

5. CONCLUSIONS

To discard the traditional TCP/IP protocol stack and program Ethernet adapter hardware directly, LLCE bypasses influence of operating system on protocol stack, and has been practiced improving on network communication latency performance and bandwidth performance. It is helpful for Ethernet to apply to some kinds of multiprocessor, distributed system and PC cluster.

However, It's found that not only 1000M TCP/IP communication but also 1000M LLCE over 1000M Intel PRO/1000 XF Ethernet card are still improvable on bandwidth since their practical bandwidths have difference from their physical bandwidth 1000M/s. In order to achieve better communication performance, a network data co-operator will be joined 1000M Intel PRO/1000 XF Ethernet adapter in the next research step. It may enhance parallel capability of host and network. In other words, by enhancing parallel capability of computing and communicating, the performance of network adapter will be improved further.

6. REFERENCES:

[1] Matt Welsh, Anindya Basu, and Thorsten von Eicken.

Low-latency communication over fast Ethernet. In Proceedings of Euro-Par, Lyon, France, August 1996, 27~29

- [2] T. von Eicken, A. Basu, and V. Buch. Low-latency communication over ATM networks using Active Messages. *IEEE Micro*, 1995, 15(1): 46~53
- [3] Prylli L, Tourancheau B. BIP: A new protocol designed for high-performance networking on myrinet. In: Rolim JDP, ed. In 1st Workshop on Personal Computer based Networks Of Workstations (PC-NOW '98) volume 1388 of Lecture Notes in Computer Science, Orlando: Springer-Verlag, 1998, 472~485
- [4] Chen ZH, Ma J, Chen GL, Gao F. Study on IP supporting over user-level protocol BCL-3. *Journal of Software(in Chinese)*, 2003,14(9):1629~1634
- [5] Shen J, Zheng WM, Ju DP. FMP: A fast messages passing for workstation clusters. *Chinese Journal of Computers*, 1998,21 (7):595~602
- [6] David Clark, Van Jacobson, John Romkey, and Howard Salwen. An analysis of TCP processing overhead. *IEEE Communications Magazine*, 1989,27(6): 23-39
- [7] Vijay Karamcheti and Andrew Chen. Software overhead in messaging layers: Where does the time go? In Proceedings of the Sixth Symposium on Architectural support for Programming Languages and Operating Systems, San Jose, California, October 1994, 51~60
- [8] Kay J, Pasquale J. The importance of non-data touching processing overheads in TCP/IP. *ACM Sigcomm. Computer Communication Review*, 1993,23(4): 259~268
- [9] Jonathan Kay and Joseph Pasquale. A performance analysis of TCP/IP and UDP/IP networking software for the DECstation 5000. Technical report, Department of Computer Science and Engineering: University of California at San Diego, 1992



Dai XinFa born in 1974 is a PH.D candidate in computer science in Harbin Engineering University, China. After receiving master degree, he joined Wuhan Digital Engineering Institute, China in 1999, where he currently is a software engineer. His main research interests are in the area of fault-tolerance and distributed computing. He has published several research papers in the topics of synchronous problem on fault-tolerant computer system, real-time and fault-tolerance on distributed computing system.



Yuan Youguang born in 1941 is a full professor, Ph.D. supervisor as with a senior member of IEEE Computer Society. He received Bachelor degree in wireless science and Master degree in computer science from Chongqing University, China in 1965 and 1982. He was the head of several national advanced research projects and was the holder of several national science and technology awards. He has published 2 works and over 150 papers. His main research interests include fault-tolerant computing, distributed computing and reliability theory.

A QoS Multicast Routing Algorithm Working with Imprecise State Information

Yan Xin¹, Li Layuan¹, Zhang Xiaoxing²

¹Department of Computer Science, Wuhan University of Technology
Wuhan, Hubei 430063, P.R. China

²Key Laboratory of High Voltage and Electrical New Technology of Ministry of Education
Chongqing University, Chongqing 400044, P.R. China
Email: yanxin@mail.whut.edu.cn Tel: +86 (27) 86533510

ABSTRACT

In large networks, maintaining precise global network state information is almost impossible. Many factors, including non-negligible propagation delay, infrequent link state update due to overhead concerns, link state update policy, and hierarchical topology aggregation, have impacts on the precision of the network state information. The existing QoS multicast routing algorithms do not provide satisfactory performance with imprecise state information. In the paper, we propose a distributed QoS multicast routing scheme based on traffic lights, called QMRI algorithm, which can probe multiple feasible tree branches, and select the optimal or near-optimal branch through the *UR* or *TL* mode for constructing a multicast tree with QoS guarantees if it exists. The proposed algorithm considers not only the QoS requirements but also the cost optimality of the multicast tree. Extensive simulations show that our algorithm achieves high call-admission ratio and low-cost multicast trees with modest message overhead. The algorithm can tolerate high degree of state information imprecision.

Keywords: QoS, Multicast routing, Imprecise state information, Traffic lights, Simulation

1. INTRODUCTION

Network routing consists of two basic tasks. The first task is to collect the network state information and keep it up-to-date. The second task is to find a satisfactory path for a new connection based on the collected information. Most published routing algorithms require every node to maintain a network state either by a multicast routing protocol or by a unicast routing protocol. However, such state information is inherently imprecise in a dynamic network where the traffic load changes constantly. The imprecision is especially noticeable in large wide-area networks due to the following three reasons. Firstly, it takes non-negligible propagation delay for a local state change to be broadcasted to other nodes. Secondly, a multicast routing or unicast routing protocol updates the state information periodically or upon triggering when significant state change is detected. There exists a tradeoff between the update frequency and the overhead involved (for instance, the usage of *hold-down timers*). For large-scale networks, the excessive communication overhead often makes it impractical for the update frequency to be high enough to cope with the dynamics of network parameters such as bandwidth and delay. Thirdly, the hierarchical approach is likely to be used to solve the scalability problem of routing in large-scale networks. However, the state aggregation in hierarchical routing increases the level of imprecision.

With the rapid increase of the requirements of multimedia and real-time services in Internet, QoS multicast routing has continued to be a very important issue in the area of network research. The main objective of multicast routing and its QoS extension is to construct a multicast tree that optimizes certain objective function (e.g., making efficient use of network resources) with respect to performance-related constraints (e.g., end-to-end delay bound, inter-receiver delay-jitter bound, minimal bandwidth available and maximal ratio of packet-loss, etc.).

In this paper, we propose a distributed QoS multicast routing scheme based on traffic lights, called QMRI algorithm, which can probe multiple feasible tree branches, and select the optimal or near-optimal branch through the *UR* or *TL* mode for constructing a multicast tree with QoS guarantees if it exists. The scheme is designed to work with imprecise state information.

2. RELATED WORK

The traditional multicast routing protocols, e.g., DVMRP and PIM, were designed for the best-effort data traffic. They construct multicast trees primarily based on connectivity. Such trees may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms provide heuristic solutions to the NP-complete problem of constrained Steiner tree, which is to find the delay-constrained least-cost multicast trees. However, these algorithms are impractical in the Internet environment because they have excessive computation overhead, require the knowledge about the global network state, and can't tackle dynamic group members. X. Jia's algorithm is a heuristic distributed scheme based on the greedy Steiner tree [1]. However, this algorithm requires excessive message processing overhead. The spanning joining protocol proposed by Carlberg and Crowcroft is able to tackle dynamic group members and does not require any global network state information [2]. However, it has excessive communication and message processing overhead because it relies on the flooding method to find a feasible tree branch to connect up a new member. Nonetheless, the algorithms mentioned above can't work with imprecise network state information efficiently.

The imprecise network state information kept at each node imposes difficulties in QoS provision. R. Guerin and A. Orda investigated the problem of QoS unicast routing, and expressed it in certain probabilistic manner when the state information is imprecise or inaccurate [3]. Their major objective was to identify a path that is mostly likely to satisfy the delay requirement, which they achieved by decomposing the end-to-end constraint into local delay constraints and

deriving tractable, near-optimal solutions for some certain classes of probability distributions. D.H. Lorenz and A. Orda further investigated the same problem, considered not only the delay requirement but also the minimal cost of the routing path [4]. S. Chen and K. Nahrstedt assumed a simplified probability model in which the end-to-end delay on the routing path is uniformly distributed, and proposed a distributed routing scheme, called ticket-based probing [5].

As mentioned above, there're some theoretical researches on either QoS multicast routing or QoS unicast routing with imprecise state information having been done. However, the theoretical research on QoS multicast routing with imprecise state information is done much less. And that, a practical QoS multicast routing algorithm working with imprecise state information has NOT been seen up to now yet.

3. SYSTEM MODELS

Network Model

A network is usually represented as a weighted, connected digraph $G=(V,E)$, where V denotes the set of nodes and E denotes the set of full-duplex, directed communication links connecting the nodes. $|V|$ and $|E|$ denote the number of nodes and links in the network, respectively. Without loss of generality, only simple digraphs are considered, in which there exists at most one link between a pair of ordered nodes.

Suppose that $s \in V$ is the source node of a multicast tree, and $M \subseteq \{V-\{s\}\}$ is a set of end nodes of the multicast tree. Let R^+ be the set of positive real numbers. For any link $e \in E$, we can define the link state information (link QoS metrics): the bandwidth function $bandwidth(e): E \rightarrow R^+$, the delay function $delay(e): E \rightarrow R^+$, and the cost function $cost(e): E \rightarrow R^+$. Similarly, for any node $n \in V$, we can define the node state information (node QoS metrics): the delay function $delay(n): V \rightarrow R^+$, and the cost function $cost(n): V \rightarrow R^+$. We also use $T(s,M)$ to denote a multicast tree, which has the following relations.

$$\begin{aligned} bandwidth(p(s,t)) &= \min\{bandwidth(e), e \in p(s,t)\} \\ delay(p(s,t)) &= \sum_{e \in p(s,t)} delay(e) + \sum_{e \in p(s,t)} delay(n) \\ cost(T(s,M)) &= \sum_{e \in T(s,M)} cost(e) + \sum_{e \in T(s,M)} cost(n) \end{aligned}$$

where $p(s,t)$ denotes the path from the source s to the end node t of $T(s,M)$.

As mentioned earlier, the network state information available for making multicast routing decisions is inherently imprecise in a dynamic network. We can express the imprecise state information by making use of certain probabilistic manner [3]. That is, we wish to find the path that is most likely to be able to accommodate the requests of a new connection and that has the minimal cost. With QoS requirements, the problem can be represented as finding a feasible and optimal (or near-optimal) multicast path p^* , such that, (1) $\prod_{e \in p^*} P_e(b) \geq \prod_{e \in p} P_e(b)$, where $e \in E$ is each link on the path and $P_e(b)$ is the probability that link e can accommodate a flow which requires b units of bandwidth, (2) $\prod_{e \in p^*} P_e(d) \geq \prod_{e \in p} P_e(d)$, where $P_e(d)$ is the probability that the delay of link e is less than d , and (3) minimizing the cost of $T(s,M)$. QoS multicast routing problem for imprecise state parameters is a NP-complete problem. When there are multiple feasible paths, we want to

select the one with the least cost [6].

Imprecise State Model

Network state information is required to be maintained at every node for every possible destination. The information is updated periodically either by a unicast routing protocol or a multicast routing protocol. It includes: (1) the up-to-date topology information of networks, (2) link state information, which consists of the residual bandwidth on a link, the propagation delay along the link, and the link cost, and (3) node state information, which consists of the queuing delay at a node and the node cost.

For the purpose of simplicity, we do not apply the imprecision model on network topology information, propagation delay and cost metric. Such a simplification will not degrade the routing performance significantly because of the following reasons. (1) Network topology can change, but is relatively infrequent comparing to the QoS state such as delay metric. (2) The propagation delay along a link is determined by Euclidean distance of the link. Its variety is utmost tiny, and is negligible [7]. (3) The cost metric is used for optimization, in contrast to the bandwidth and delay metrics used in QoS constraints. Since there is not a strict cost bound requirement, certain degree of imprecision for the cost metric is tolerable.

In order to capture the imprecision of the state information induced by the dynamic change of network state, $\forall (i, j) \in E$, we can use $\Delta b(i, j)$ to denote the *estimated* maximum change of link bandwidth $b(i, j)$ before the next update. That is, based on the recent state history, the actual bandwidth of link (i, j) is expected to be between $b(i, j) - \Delta b(i, j)$ and $b(i, j) + \Delta b(i, j)$ in the next update period. We'll try to calculate $\Delta b(i, j)$ through the following way. Let $b_{old}(i, j)$ and $b_{new}(i, j)$ be the values of $b(i, j)$ before and after the update, respectively. Similarly, let $b_{old}(i, j)$ and $b_{new}(i, j)$ be the values of $b(i, j)$ before and after the update, respectively. $b_{new}(i, j)$ is provided by a distance-vector protocol, $b_{new}(i, j)$ is calculated as follows.

$$\begin{aligned} \Delta b_{new}(i, j) &= \alpha \times \Delta b_{old}(i, j) \\ &+ (1 - \alpha) \times |b_{new}(i, j) - b_{old}(i, j)| \end{aligned} \quad (1)$$

The factor $\alpha (\alpha < 1)$ determines how fast the history information ($b_{old}(i, j)$) is forgotten, and $(1 - \alpha)$ determines how fast $b_{new}(i, j)$ converges to $|b_{new}(i, j) - b_{old}(i, j)|$.

For the imprecision of the end-to-end delay along a path $p(s, t)$, we use $\Delta d(s, t)$ to denote the *estimated* maximum change of the end-to-end delay along the path $(d(s, t))$ before the next update [5]. According to the above assumption, the actual delay of the path is also between $d(s, t) - \Delta d(s, t)$ and $d(s, t) + \Delta d(s, t)$ in the next update period. Similarly, $\Delta d_{old}(s, t)$ and $\Delta d_{new}(s, t)$ are the values of $\Delta d(s, t)$ before and after the update. $d_{old}(s, t)$ and $d_{new}(s, t)$ are the values of $d(s, t)$ before and after the update. The value of $\Delta d_{new}(s, t)$ is

$$\Delta d_{new}(s, t) = \beta \times \Delta d_{old}(s, t) + (1 - \beta) \times |d_{new}(s, t) - d_{old}(s, t)| \quad (2)$$

Similarly, the factor $\beta (\beta < 1)$ determines how fast the history information ($d_{old}(s, t)$) is forgotten, and $(1 - \beta)$ determines how fast $d_{new}(s, t)$ converges to $|d_{new}(s, t) - d_{old}(s, t)|$.

We further assume that the bandwidth on link (i, j) and the end-to-end delay along path $p(s, t)$ are uniformly distributed on $[b_{new}(i, j) - \Delta b_{new}(i, j), b_{new}(i, j) + \Delta b_{new}(i, j)]$ and $[d_{new}(s, t) - \Delta d_{new}(s, t), d_{new}(s, t) + \Delta d_{new}(s, t)]$, respectively, and that, B and D denote the minimum bandwidth requirement of each link on path $p(s, t)$ and the end-to-end delay constraint along the path, respectively. Therefore, we can calculate the probability which path $p(s, t)$ satisfies the bandwidth and delay requirements.

$$P = mP(b(s, t) \geq B) + nP(d(s, t) \leq D) \quad (3)$$

The factors, m and n determine the concern degrees about the bandwidth requirement and the delay constraint, respectively. In Eq. (3), $P(b(s, t) \geq B)$ and $P(d(s, t) \leq D)$ are calculated as follows.

- 1) If $\forall (i, j) \in p(s, t), B > b_{new}(i, j) + \Delta b_{new}(i, j)$, then $P(b(s, t) \geq B) = 0$. Otherwise,

$$P(b(s, t) \geq B) = \prod_{(i, j) \in p(s, t)} P(b(i, j) \geq B) = \prod_{(i, j) \in p(s, t)} \frac{\min\{b_{new}(i, j) + \Delta b_{new}(i, j) - B, 2\Delta b_{new}(i, j)\}}{2\Delta b_{new}(i, j)}$$

- 2) If $D < d_{new}(s, t) - \Delta d_{new}(s, t)$, then $P(d(s, t) \leq D) = 0$. Otherwise,

$$P(d(s, t) \leq D) = \frac{\min\{D - (d_{new}(s, t) - \Delta d_{new}(s, t)), 2\Delta d_{new}(s, t)\}}{2\Delta d_{new}(s, t)}$$

4. PROPOSED ALGORITHM

Overview

The proposed algorithm, called QMRI algorithm, should operate on top of some unicast routing protocol such as distance-vector protocol that can pre-compute the paths, which the algorithm needs. The QMRI algorithm constructs a multicast tree that can meet the bandwidth and delay requirements, as well as, either the *optimal cost* or the *maximum probability* of satisfying the bandwidth and relay requirements. Our scheme based on *traffic lights* can probe multiple feasible paths in parallel through two routing modes, and make the routing decisions by watching the colors of the control messages, just like watching the traffic lights on a cross road.

The control messages of the QMRI algorithm are defined as follows. (1) Join-Request (Join-Req) is a probe message of joining request sent towards the source node of a multicast group by a new member that wants to join in them. The message can accumulate the additive metric such as delay, and memory the concave metric such as bottleneck bandwidth of the path it searches. The metrics are used to calculate the state of the path and to see if it satisfies the QoS requirements. (2) Reply-Green (Reply-G) is an acceptance reply sent

downstream towards the new member by the source that accepts the new member's joining request. That means some path can satisfy the bandwidth and delay requirements of the joining request *completely*. The path is called a feasible path that enables the value of Eq. (3) to be 1. (3) Reply-Yellow (Reply-Y) is a pre-alert message sent downstream towards the new member by some node to which the path is only able to *partially* satisfy the bandwidth and delay requirements of the joining request. The path is called a possibly feasible path that enables the value of Eq. (3) to be between 0 and 1. (4) Reply-Red (Reply-R) is a rejection message sent downstream towards the new member by some node that rejects the joining request. That is, the path to the node *can't* satisfy the bandwidth and delay requirements completely (the value of Eq. (3) is 0). Reply-Y and Reply-R can enable the immediate downstream node of some node to enter the *TL* (traffic lights) mode. Each type of message on some path can carry the value of the probability which the path satisfies the bandwidth and delay requirements.

In the QMRI algorithm, a multicast member can join or leave a multicast session dynamically. Thus, It is very important that the process of a multicast member's joining/leaving should not disrupt or interfere with the ongoing multicast session. For that, we use a method that a multicast tree is formed in an *incremental* manner in order to implement the seamless transformation of the multicast tree. In addition, when a receiver leaves a multicast session, it should send a leaving message upstream along the on-tree branches to a fork node. After receiving the leaving message, the intermediate node (fork node) will release the corresponding network resources. The rest of the multicast tree remains unchanged.

Detailed Description

In this section, we will describe the details of the QMRI algorithm by an illustration shown in Figure 1, in which the number beside each link is the cost of the link.

In our scheme, the multicast tree is formed in an incremental manner. When a new member t intends to join in a multicast session, it sends a probe message for the joining (Join-Req) towards the source s of the session, and initializes the routing process in terms of the unicast routing (*UR*) mode. The *UR* mode is a routing mode by which the QMRI algorithm searches for the shortest (i.e., the *minimal cost* between t and s) path through a *Bellman-Ford* algorithm or a *Dijkstra* algorithm. When an intermediate node u that has been an on-tree node receives the Join-Req message, it will make an eligibility test. The node u will check whether the existing QoS guarantees on the path between u and t meet the probability requirements of the bandwidth and relay constraints of the new member or not. That is, node u will see if the value of Eq. (3) is 1.

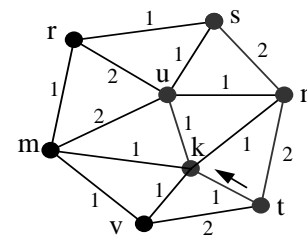


Figure 1. Network illustration

Suppose that $b(t, k)$ and $d(t, k)$ are the bandwidth and delay of

the path from t to k , respectively. Similarly, $b(k,u)$ and $d(k,u)$ denote the bandwidth and delay of path (k,u) , respectively. Recall that the bandwidth and delay constraints are B and D , respectively. According to Eq. (1) and Eq. (2), node u will calculate and check whether

$$(b(u,k) + \Delta b(u,k) \geq B) \wedge (b(k,t) + \Delta b(k,t) \geq B) \wedge (d(u,t) + \Delta d(u,t) \leq D) \quad (4)$$

If the expression is true (i.e., the value of the probability is 1), the Join-Req message is continuously sent forward to the immediate upstream node of node u . This process will be repeated until the source s receives the Join-Req message eventually. If source s verifies the similar test, it will send a Reply-G message to the new member. And then, the joining connection is established. If Expression (4) is false (i.e., the value of the probability is 0 or in the range (0,1)), node u will transmit either a Reply-Y message or a Reply-R message to its immediate downstream node k . There exist two cases needed discussing. (1) If $(b(u,k) - \Delta b(u,k) < B) \vee (b(k,t) - \Delta b(k,t) < B) \vee (d(u,t) - \Delta d(u,t) < D)$ is true (i.e., the value of the probability is 0), a Reply-R message from node u will be send to node k . (2) Otherwise (i.e., the value of the probability is in the range (0,1)), node k will receive a Reply-Y message that carries the value of the probability which path (u,t) satisfies the bandwidth and delay requirements. The value of the probability can be calculated according to Eq. (3).

At the moment, node k enters into the fork routing mode based on traffic lights (TL). Under the TL mode, the QMRI algorithm may probe another several paths in the same method as mentioned above. For these probed paths, node k will make the routing decisions according to the following four cases in order.

1) If node k receives more than one Reply-G messages, it will further check these feasible paths, i.e. the path associated with the Reply-G message, to see if

$$(b(*,k) + \Delta b(*,k) \geq B) \wedge (b(k,t) + \Delta b(k,t) \geq B) \wedge (d(*,t) + \Delta d(*,t) \leq D) \wedge \text{cost}(*,t) = \min[\dots] \quad (5)$$

Among multiple feasible paths, that one that can make Expression (5) is true is just the optimal (minimal cost) or near-optimal routing path for connecting up the new member t to the multicast session.

2) If node k receives a single Reply-G message, it selects the only path carrying the Reply-G message as the optimal or near-optimal routing path.

3) If all or a portion of the messages which node k received are the Reply-Y messages, and that no Reply-G message in them, node k will select the path from which the Reply-Y message has the *maximum* value of the probability which the path satisfies the bandwidth and delay constraints. Recall that the path associated with the Reply-Y message is called a possibly feasible path.

4) If all of the messages which node k received are the Reply-R messages, node k will incorporate the messages into one, and further send it to the immediate downstream node of node k . The above whole process including the UR mode and the TL mode will be repeated until the new member t eventually receives the Reply-G or Reply-R message.

Seen from the above steps, in our scheme, the intermediate nodes in a multicast tree can generally do the route computation in a *distributed* manner. The QMRI algorithm is mainly suitable to the intra-domain operating environment.

Analysis of Complexity

The computation complexity and the number of messages needed to construct a multicast tree are two main factors of impacting the complexity of the QoS multicast routing algorithm.

The computation complexity of the QMRI algorithm only depends on the unicast protocol, if the joining path is computed by on-demand routing. At present, the computation complexity of the QoS routing heuristics with two QoS metrics (delay and bandwidth) is $O(|V||E|)$, where $|V|$ is the number of nodes and $|E|$ is the number of edges in a network. For most networks, $|E|=O(|V|)$, hence the complexity is $O(|V|^2)$ [8]. For a multicast group with $|M|$ members, the computation cost is $|V|^2|M|$. Thus, the computation complexity of the QMRI algorithm is $O(|V|^2|M|)$.

For the message exchange, the QMRI algorithm mainly deals with four types of message: Join-Req, Reply-G, Reply-Y and Reply-R. However, in each step of the computations, the algorithm deals with *at most* three types of message. This means that a multicast group with $|M|$ members should deal with $3|M|$ messages. A Join-Req message is processed by *at most* $|E|$ hops along the way up to the node where it is accepted or rejected. The overhead of the message processing for joining $|M|$ receivers is $3|ME|$ at most. Thus, the message complexity of the QMRI algorithm is $O(3|ME|)$.

5. SIMULATIONS

Extensive simulations were done to evaluate the performance of the proposed QMRI algorithm. The simulations are implemented by using NS2, which has been developed to be able to simulate the imprecise network state information [9]. Two performance metrics, success ratio and network cost, are defined as follows.

$$SR = \frac{Ns}{Nr}, \quad NC = Rn$$

where Ns is the number of joining requests accepted, and Nr is the total number of joining requests. The “joining requests accepted” means the established multicast tree satisfies the QoS constraints. Network cost NC that consists of routing cost and message overhead is measured by the average value of the total number of simulation runs that is denoted by Rn . Three algorithms are simulated: the flooding algorithm, the PIM-SM algorithm, and the QMRI algorithm. All of the experiment results are the average values obtained by simulating time after time.

The network topology used in our simulations is generated by Waxman’s algorithm [10], which has 50 nodes. The source node and the destination nodes of a multicast tree are randomly generated. In order to simulate the real situations, group size G (i.e., the multicast membership) is always made less than 20% of the total nodes.

In the experiments, the bandwidth $b(i,j)$ and the delay $d(i,j)$ of each link (i,j) are uniformly distributed in the range of [30Mbps, 45Mbps] and the range of [5ms, 50ms], respectively.

The bandwidth requirement B and the end-to-end delay requirement D of each connection request are uniformly distributed in the range of [10Mbps, 60Mbps] and the range of [30ms, 160ms], respectively. The cost of each link is uniformly distributed in the range of [1, 200]. All of the experiment values are randomly generated in their distribution ranges. Each link (i, j) is associated with two bandwidth values: $b(i, j)$ and $b\text{-new}(i, j)$. $b(i, j)$ is the bandwidth value of the link used to compute routing. $b\text{-new}(i, j)$ is the actual bandwidth of the link at the time of routing. $b\text{-new}(i, j)$ is uniformly distributed in $[(1 - \lambda)b(i, j), (1 + \lambda)b(i, j)]$, where λ is a simulation parameter, called imprecision rate, specifying the largest percent difference of $b\text{-new}(i, j)$ from $b(i, j)$.

$$\lambda = \sup_{i, j} \text{remum} \left\{ \frac{|b - \text{new}(i, j) - b(i, j)|}{b(i, j)} \right\}$$

Similarly, we can also define the delay imprecision rate of each link, simulation parameter λ . For the purpose of simplicity, let $\lambda = 10\%$ in our experiments.

Success Ratio

Figures 2-3 compare the success ratios of the three algorithms. The success ratio is a function of bandwidth requirement B , end-to-end delay requirement D , and imprecision rate λ . Figures 2-3 show the following results.

(1) The flooding algorithm, as expected, has the best success ratio, and is almost irrelevant to the imprecision rate. (2) The success of the QMRI algorithm is very close to that of the flooding algorithm, even when the imprecision rate is as high as 50%. (3) The PIM-SM algorithm performs much worse when the imprecision rate is high.

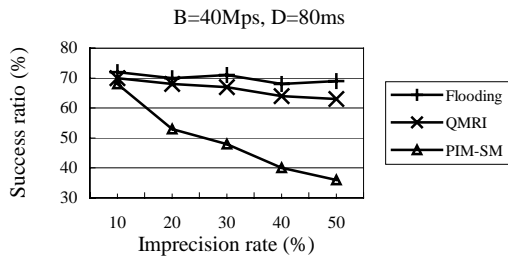


Figure 2. Success ratio ($B=40\text{Mbps}$, $D=80\text{ms}$)

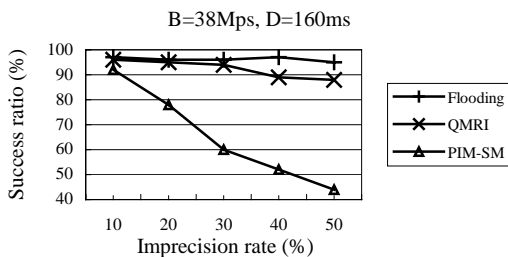


Figure 3. Success ratio ($B=38\text{Mbps}$, $D=160\text{ms}$)

Network Cost

The network cost is simulated against end-to-end delay constraint D . During this round of simulations, the multicast group size is 4. At each simulation point, the simulation runs 100 times. We define the smallest value of D as $d_{\max} = \max\{d(s, t) \mid \forall (s, t) \in G, d(s, t) \text{ is the delay on the shortest path from the source } s \text{ to the destination } t\}$. D starts from d_{\max} and is increased by $d_{\max}/8$ every time. The increment of $d_{\max}/8$

is selected to capture the utmost of the tendency of network cost against the change of D after many simulation runs. Since for each simulation run, G is different, thus d_{\max} is also different. The values of D on the x-axis are the average values of D in all runs.

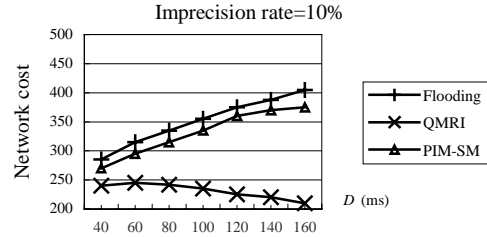


Figure 4. Network cost ($\lambda=10\%$)

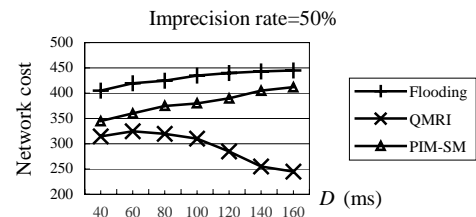


Figure 5. Network cost ($\lambda=50\%$)

From figures 4-5, it can be seen that the QMRI algorithm has a much lower network cost than the flooding algorithm and PIM-SM. This is because the QMRI algorithm uses not only the bandwidth metric and the delay metric but also the cost metric to make the routing decision while the other two algorithms use only the bandwidth metric or the delay metric. We can also see that the network cost of the QMRI algorithm decreases as the delay constraint is relaxed.

6. CONCLUSIONS

In this paper, we proposed a distributed QoS multicast routing scheme based on traffic lights, which works for dynamic networks where the network state information maintained at every node is imprecise. Our simulations show that the scheme achieves high call-admission ratio and low-cost multicast trees with modest message overhead. It can tolerate high degree of state information imprecision.

Our ongoing work includes: (1) a research on how different state update policies affect the routing performance of the proposed scheme and how to choose the state update policies, (2) a probe of the impact on the path selection process, of the imprecise state information induced by the aggregation process that occurs in hierarchically interconnected networks, (3) a further investigation on the proposed algorithm's suitability for its using in an inter-domain multicast and hierarchical network environment.

7. REFERENCES

- [1] X. Jia, "A Distributed Algorithm of Delay-Bounded Multicast Routing for Multimedia Applications in Wide Area Networks", IEEE/ACM Transactions on Networking, Vol.6, No.6, Dec. 1998, pp.828-837.

- [2] K. Carberg, J. Crowcroft, "Building Shared Trees Using a One-to-Many Joining Mechanism", ACM Computer Communication Review, Jan. 1997, pp.5~11.
- [3] R.A. Guerin, A. Orda, "QoS Routing in Networks with Inaccurate Information: Theory and Algorithms", IEEE/ACM Trans. on Networking, Vol.7, No.3, June 1999, pp.350~364.
- [4] D.H. Lorenz, A. Orda, "QoS Routing in Networks with Uncertain Parameters, IEEE/ACM Trans. on Networking", Vol.6, No.6, Dec. 1998, pp.768~778.
- [5] S. Chen, K. Nahrstedt, "Distributed QoS Routing with Imprecise State Information", in Proc. of ICCCN'98, 1998.
- [6] G. Apostolopoulos, R.A. Guerin, S. Kamat and S. Tripathi, "Improving QoS Routing Performance under Inaccurate Link State Information", in Proc. of ITC'16, June 1999, pp.1351~1362.
- [7] Sun, H. Langendorfer, "A New Distributed Routing Algorithm for Supporting Delay-Sensitive Applications", Computer Communications, Vol.21, No.6, 1998, pp.572~578.
- [8] Li Layuan, Li Chunlin, "A Multicast Routing Protocol with Multiple QoS Constraints", in Proc. of WCC, Aug. 2002.
- [9] J. Rexford, A. Shaikh, "Performance Evaluation of Quality-of-Service Routing with Inaccurate Link State Information", in Proc. of OPENSIG Fall'97 Workshop, Columbia University, October 1997.
- [10] B.M. Waxman, "Routing of Multiple Connections", IEEE Journal on Selected Areas in Communications, Vol.6, No.9, 1998, pp.1617~1622.



Yan Xin was born in 1970. He is a Ph.D. candidate in Department of Computer Science, Wuhan University of Technology. His research interests include high-performance computer networks and network simulation technology.



Li Layuan was born in 1946. He is a professor and Ph.D. tutor in Wuhan University of Technology. His research interests include high-speed computer networks, protocol engineering and image processing.

Data Transmission Rate Control in Computer Networks Using Neural Predictive Networks*

Liansheng Tan, Naixue Xiong and Yan Yang

Department of Computer Science, Central China Normal University

Wuhan, Hubei province, PR China

Email: L.Tan@mail.ccnu.edu.cn Tel: 0086-27-67867651

ABSTRACT

In this paper, a novel congestion control scheme is proposed which is based on a Back Propagation (BP) neural network method. The BP neural network predicts the dynamic buffer occupancy of the bottleneck node. The proposed control scheme avoids congestion efficiently and optimizes the transmission performance as shown by the theoretic analysis and simulation results.

Keywords: BP neural network, congestion control, data transmission, computer network.

1. INTRODUCTION

With the rapid development of computer networks, more and more severe congestion problems have occurred. Designing efficient congestion control scheme is, therefore, a crucial issue to alleviate network congestion and to fulfill data transmission effectively. The main difficulty in designing such scheme lies in the large propagation delay in transmission that usually leads to a mismatch between the network resources and the amount of admitted traffic. To overcome this difficulty, ref. [1] suggests using fuzzy control to realize the rate-based network congestion control, and the generic algorithm in queue strategy is presented in [2-3]. Furthermore, [4-5] use a multi-step neural predictive technique to predict the congestion situation in computer networks, but the longer predictive steps has still existed and the effectiveness is greatly limited in existed papers. And yet the responsiveness of the congestion control scheme is crucial to the stability of the whole network system and the relevant performance, this issue is, however, not considered in these works.

To overcome the delay related difficulty in designing an efficient congestion control algorithm; in this paper we suggest a novel congestion control scheme on the basis of the BP neural network. In this scheme, the BP neural predictive controller is suggested to be located at the sources rather than at the switch in order to improve the predictive scheme. This is due to the fact the less prediction horizon usually leads to better accuracy, whereas in the proposed scheme the predictive horizon is linked with the network structure. Therefore, the method usually brings forth better performance in terms of predictive accuracy and efficiency. Thereby, the proposed scheme can response the network change quickly and avoid the network congestion effectively.

2. CONGESTION CONTROL MODEL

In this section, we consider a general model as shown in Fig. 1, where different connections and various traffic requirements are mapped into different classes. For the i th source, at time-slot n the rate control algorithm computes the low priority bandwidth $i_L(n)$, which is left over by the sum of the highest priority traffic $i_{H1}(n)$ and the higher priority traffic $i_{H2}(n)$. Let $i(n)$ denote the sending rate of this source we then have $i(n) = i_L(n) + i_{H1}(n) + i_{H2}(n)$. The component $x_{iL}(n)$ is the number of $i_L(n)$ packets waiting in the queue. The component $x_{i0}(n)$ is the queue threshold and is assumed to be a constant [4-5].

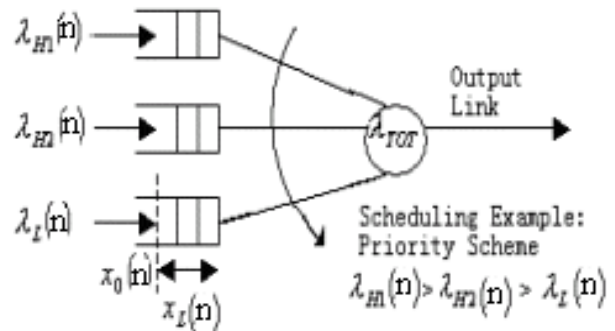


Fig.1 A simple model with one source

The model of multiple sources and a bottleneck with controllers is shown in Fig. 2. We still assume the model has N sources although the number of active sources denoted by N_0 may vary with time. The switching node has a finite buffer size K to store the control packets (CPs) and data packets, and can send such packet at a constant service rate μ .

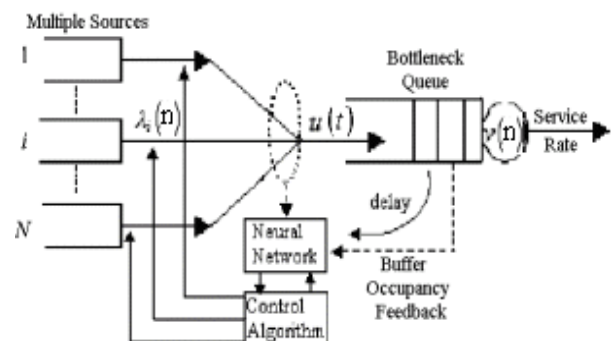


Fig.2 The model with multiple sources and a bottleneck implemented by a neural network controller.

*This research is supported by the National Natural Science Foundation of China under Grant No. 60174043 and the Key Project of Natural Science Foundation of Hubei Province of China under Grant No. 2002AB025.

The key component of this model is the neural network and the control algorithm. The procedure to be implemented is specified as follows: each source sends data to the bottleneck

node at fixed intervals, and the bottleneck node feedbacks the relevant control information to each source. According to this feedback information, the BP neural network located at the sources predicts the dynamic buffer occupancy of the bottleneck node. The controlled best-effort traffic of the sources just uses the bandwidth, which is left by the guaranteed traffic for the later has the higher priority compared to the best-effort traffic. The sources adjust their sending rates according to their available resources. Therefore, the dynamic system of a switching node in a network can be described by the following non-linear time-variant and time-delayed equation [6-7].

$$x(n+1) = \text{Sat}_K \left\{ x(n) + \sum_{i=1}^N e_i \lambda_i(n - \tau_{li}) - \mu \right\} \quad (1)$$

where we assume that the signals are sampled every T msec, the component τ_{li} is the forward delay from the i th source node to the switching node, $x(n)$ is the buffer occupancy at time slot n , and

$$e_i = \begin{cases} 1, & \text{source } i \text{ is active;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\text{Sat}_K \{x\} = \begin{cases} K, & x > K; \\ x, & 0 \leq x \leq K; \\ 0, & x < 0; \end{cases}$$

Here, we also assume the following values: The virtual connection (VCs) delay d_i equals the forward delay τ_{li} plus its corresponding backward delay τ_{2i} from the switching node to the i th source. The round trip delay d is set to be $d = \min(d_1, d_2, \dots, d_N)$, and the input delay. The minimum delay τ_1 is set to be $\tau_1 = \min(\tau_{11}, \tau_{12}, \dots, \tau_{1N})$. The low priority traffic can only be transmitted when no congestion appears in the network. Furthermore, we assume that the service is first-come-first-served (FCFS) and the packet length is constant. The buffer occupancy $x(n)$ at time slot n is measured, the CPs are feed-backed to the controlled sources every T seconds.

3. THE PREDICTIVE CONTROL TECHNIQUE

The BP neural network algorithm is introduced into this paper as a predictive mechanism. We assume the number of input neuron is N , and the number of sample study group is M . The sample study groups are independent from each other. We further assume the output of the study sample group (teaching assigns) is $R_j^{(k)}$ ($j \in [0, N]$, $k \in [1, M]$) and the actual output for output element j in the network is $O_j^{(k)}$. So $E^{(k)}$ is set to be the k^{th} group input goal function. Therefore, we have $E^{(k)} = \sum_j (R_j^{(k)} - O_j^{(k)})^2 / 2$. The total goal

function is $J = \sum_k E^{(k)}$. If $J \rightarrow 0$, 0 is a constant

that is small enough and $\rho > 0$, then the algorithm is terminated; Otherwise adjust the weight W between the implicit layer and output layer until it satisfy the expected difference value.

A neural network mechanism is applied to determine how a BP neural network algorithm satisfies its data transfer requirement. As shown in Fig. 2, the BPNN predictive controller located at the sources predicts the buffer occupancy efficiently; the neural model for the above system can be expressed as:

$$\hat{x}(n+1) = \hat{f}[\hat{x}(n), \dots, \hat{x}(n-l+1), \lambda(n-\tau_1-1), \dots, \lambda(n-\tau_1-m-L)] \quad (2)$$

Where $\hat{x}(n-i)$ ($1 \leq i \leq l-1$) is the history buffer occupancy and $\lambda(n-j)$ ($i+1 \leq j \leq i+m+L$) is the history sending rate of the source j . L is the number of predictive step, $L = i+1$, and L, m is constant integer. $\hat{f}[\cdot]$ is the unknown function, which may be expressed by the neural network. The explicit mechanism of BP neural network L -step forward prediction is shown in Fig. 3, the buffer occupancy $x(n)$ and the history values (the past buffer occupancies: $x(n-1), x(n-2), \dots, x(n-l+1)$ the past source sending rates: $\lambda(n-l-1), \lambda(n-l-2), \dots, \lambda(n-l-m-L)$ are used as the known input of neural network. Every layer denotes one-step forward predictive, so $\hat{x}(n+L)$ on the output layer is the L -step prediction of $x(n)$. We can compute the expected total rate $\hat{\lambda}(n)$ of the N sources using the following equation:

$$\hat{x}(n+L) = \text{Sat}_K \{ \hat{x}(n+L-1) + \hat{\lambda}(n) - \mu \} \quad (3)$$

Based on the rate $\hat{\lambda}(n)$ the source i adjusts the sending rate $\lambda_{il}(n) = \hat{\lambda}(n) \delta_i(n) - \lambda_{ih_1}(n) - \lambda_{ih_2}(n)$, and $\delta_i(n)$ is a factor of share the available resources to source i , ($1 \leq i \leq N$, $\sum_{i=1}^N \delta_i(n) = 1$). The specific algorithm is given in

Fig. 4. At the next instant $n+1$, we can get a new actual measured value $x(n+1)$ and a new group of history measure values: $x(n), x(n-1), \dots, x(n-l+2), \lambda(n-l), \lambda(n-l-1), \dots, \lambda(n-l-m-L+1)$ which can be used as the next instant inputs of neural network. Then the buffer occupancy $\hat{x}(n+L+1)$ can be predicted.

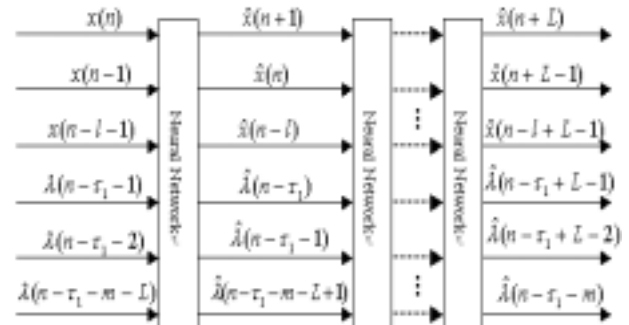


Fig. 3 The Back Propagation (BP) L -step ahead prediction, and $\hat{x}(n+L)$ is the L -step predictions of $x(n)$.


```

Procedure Neural_Network_Prediction (  $x, \lambda, \hat{x}, \hat{\lambda}$  )
{At time instant  $n$ ;
Step 1: Get the buffer occupancy  $X(n)$ , and the history
measurement value by the controller: the past buffer
occupancy:  $X(n-1), X(n-2), \dots, X(n-l+1)$ ; the past total
sending rate:  $(n-l-1), (n-l-2), \dots, (n-l-L)$ . All values above are used as available
known input of neural network.
Step 2: Predict  $L$  step  $\hat{x}(n+L)$ .
Predict  $\hat{x}(n+i), i \in [1, L]$ ;
 $[\hat{x}(n+i-1), \hat{x}(n+i-2), \dots, \hat{x}(n+i-l), \lambda(n+i-\tau_1-2), \dots,$ 
 $\lambda(n+i-\tau_1-L-m-1)]$ 
as the neural network input. These predictions are to be used to
predict the next set.
Train neural network computes the goal function  $J$  and adjust
the weight;
Back propagation for the next  $L$  step ahead prediction;
Go ahead until  $L$  step ahead predictions. Then,  $\hat{x}(n+L)$ 
is the  $L$ -step prediction.
Step 3: Compute  $\hat{\lambda}(n), \hat{\lambda}_i$  and  $\hat{\lambda}_{iL}(n)$ .
Step 4: At the next instant  $n+1$ , update the history value,  $(n+1)$ 
takes the place of  $n$ , go to step 1.
Return  $(\hat{x}(n+L) \in (\hat{x}, \hat{u}))$ .}

```

Fig. 4 Algorithm for on-line control and neural network training at sources.

4. THE SIMULATION RESULTS

To evaluate the performance of the proposed congestion control method, we focus upon the simulation model with eleven sources and one switch bottleneck node (Fig. 5), and assume that the sources always have data to transmit. The higher priority traffic, i.e., the sum of iH_1 traffic and iH_2 traffic at source i , is acquired from the real-time trace data traffic.

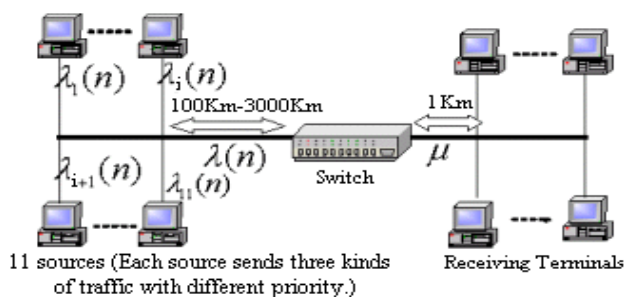


Fig. 5 The simulation model.

As shown in Fig. 5, the maximum sending rate of every source is $r_0=15.5$ Mbps. We use a simple resource sharing policy, i.e., the network bottleneck node equally shares the available bandwidth among every source. The sources start to transmit data at time $t=1$ msec together. We assume the sending rate of the switch node is $r_s=155$ Mbps, and the maximum traffic rate from the sources to the switch node is the same as the sending rate of the switch node. The sampling time interval T is 1msec and the congestion threshold is set as $x_0=1000$ Kb. We propose to use a direct multi-step neural

predictive architecture with 3-layer neural network, where in the number of the input data, the input neurons, the hidden neurons and the output neurons are all $(L+m+l)$. There are l ($l=8$) terms of buffer occupancy x and $(L+m)$ terms of the total input u . The prediction horizon is $L = l_i + 1$, and the control horizon is $N = L - l_i + 1 = 2$. To investigate the performance of this model, we set the distance from sources to switch node to be 2600Km with the forward delay $\tau_{1i} = 13$ msec, and the feedback delay $\tau_{2i} = 12$ msec, ($i=1, 2, \dots, 11$). So the RTD is $d=25$ msec. We also assume that the RTD is dominant compared to other delays such as processing delays and queuing delay, etc.

For this case, the prediction horizon is $L=14$, and $m=6$. Fig. 6 shows the higher priority ($\lambda_{H1} + \lambda_{H2}$) traffic rate. Fig. 7 shows the buffer occupancy, the predictive buffer occupancy and the actual buffer occupancy are described with broken line and real line respectively. The predictive value of the buffer occupancy is acquired at the time slot n ($n = l_i + L + 9 = 36$). Fig. 8 shows the lowest priority data rate, which is yielded based on the equation (1) and the predicted buffer occupancies from the time slot 12 to $(500 - l_i - L) = 473$. One observes that the queue size is guaranteed close the threshold of 1000Kb by the proposed neural predictive congestion control technique. The average relative error between the predictive buffer occupancy and actual buffer occupancy is $1.5e-002$, which is excellent in terms of accuracy. Fig. 9 shows total input rates.

Compared to the results in [4-5], quicker transient response of the source sending rates is acquired in our algorithm. Under the same simulation conditions, we use less predictive steps than that in [4-5]. In addition, the neural predictive controller in our method is located at the sources rather than at the switch, this usually brings forth better performance in terms of prediction accuracy and efficiency.

5. CONCLUSIONS

This paper has presented a dynamic resource management mechanism for computer communication networks on the basis of an adapting BP neural network control technique. Also we further explored the relevant theoretic foundations and the detailed implementation procedure for congestion control. Simulation results demonstrate that the proposed algorithm is excellent in system response, predictive accuracy and efficiency.

6. REFERENCES

- [1] Rose Qingyang Hu and David W. Petr, "A Predictive Self-Tuning Fuzzy-Logic Feedback Rate Controller," IEEE/ACM Transactions on Networking, December 2000, Vol. 8, No. 6, pp. 689 - 696.
- [2] Giuseppe Ascia, Vincenzo Catania, and Daniela Panno, "An efficient buffer management policy based on an integrated Fuzzy-GA approach," IEEE INFOCOM 2002, New York, June 23 - 27, 2002, No.107.
- [3] G. Ascia, V. Catania, G. Ficili and D. Panno, "A Fuzzy Buffer Management Scheme for ATM and IP Networks," IEEE INFOCOM 2001, Anchorage, Alaska, April 22-26, 2001, pp.1539-1547.
- [4] J. Aweya, D.Y. Montuno, Qi-jun Zhang and L.

- Orozco-Barbosa, "Multi-step Neural Predictive Techniques for Congestion Control –Part 2: Control Procedures," *International Journal of Parallel and Distributed Systems and Networks*, 2000, Vol. 3, No. 3, pp. 139-143.
- [5] J. Aweya, D.Y. Montuno, Qi-jun Zhang and L. Orozco-Barbosa, "Multi-step Neural Predictive Techniques for Congestion Control –Part 1: Prediction and Control Models," *International Journal of Parallel and Distributed Systems and Networks*, 2000, Vol. 3, No. 1, pp. 1-8.
- [6] L. Benmohamed and S. M. Meerkov, "Feedback Control of Congestion in Packet Switching Networks: The Case of Single Congested Node," *IEEE/ACM Transaction on Networking*, December 1993, Vol. 1, No. 6, pp. 693-708.
- [7] J. Filipiak, "Modeling and Control of Dynamic Flows in Communication Networks," Springer Verlag Hardcover, New York, May 1, 1988.

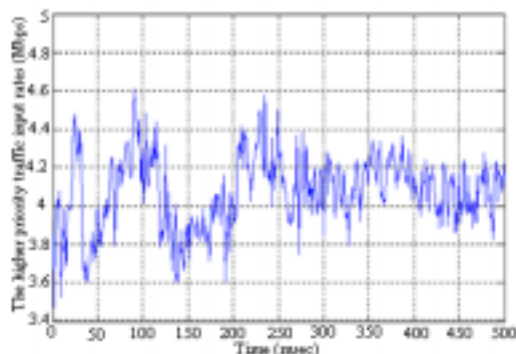


Fig. 6 The higher priority traffic rate.

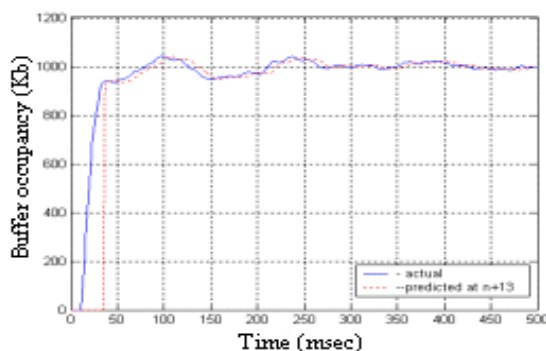


Fig. 7 The buffer occupancy.

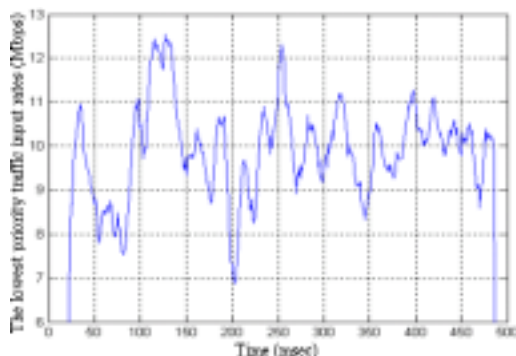


Fig. 8 The lowest priority traffic rate.

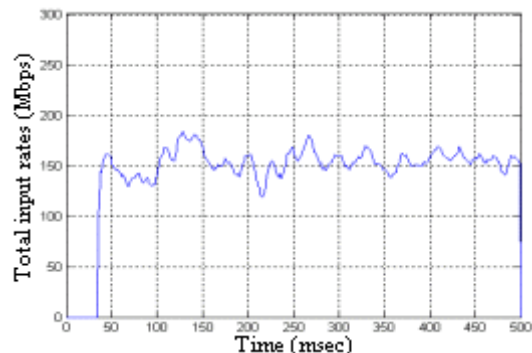


Fig. 9 The total input rates.



Liansheng Tan is now a Full Professor and Head of Department in Department of Computer Science, Central China Normal University, PR China. Professor Tan received his Ph.D. degree from Loughborough University in the UK in 1999. He was doing research in computer communication network in School of Information Technology and Engineering at University of Ottawa,

Ontario, Canada as a postdoctoral research fellow and a visiting research scientist in 2001. He has published over fifty referred papers. His research interests are in modeling, analysis and performance evaluation of computer communication networks, their protocols, services and interconnection architectures.

URL: <http://cs.ccnu.edu.cn/teacher/tls/tanliansheng.htm>.

Research on the Basis of QoS Routing Protocol of Ad Hoc Network*

Chen Niansheng^{1,2} Li Layuan¹

¹School of Computer Science, Wuhan University of Technology, Wuhan 430063, China

²Department of Computer Science, Hubei Normal University, Huangshi, 435002, China

Email: Chnsh@public.hs.hb.cn Tel: 0714-6515405

ABSTRACT

Ad Hoc network is an ideal technology to set up a fast communication system for military use and civil use. It is an important problem it faces of how to offer different QoS for them; and QoS routing technology is the key technology and a hot issue in it. This article has analyzed and studied several kinds of QoS routing algorithms that different researchers put forward. It has also compared and analyzed them from unidirectional link supports, multicast functions and QoS parameter selection. The study helps to improve and perfect the QoS routing technology of Ad Hoc network.

Keyword: Ad Hoc network, QoS routing, routing protocol

1. INTRODUCTION

Mobil Ad Hoc network^[1] is a kind of special new-typed mobile wireless network, and it is prospected to be widely used. As a kind of special form of mobile computation, it is a multihop wireless network temporarily formed by a series of dynamical host nodes. It does not have a regular base or basic network. Each node in the network serves as the host computer and route. As in the areas where the basic network does not exist, or the areas where the basic network has been destroyed, or where the basic network has been established but can not be used conveniently, the Ad Hoc network can establish a new network rapidly to realize data transmission^[2,3].

QoS (quality of service) is sets of network service demands for business flow while the network is transmitting business flow. The business flow means grouping flow that correlates with specific QoS and flows from source to destination^[4]. So QoS is a set of measurable requests that applied business demands the network transmission service. It includes bandwidth, end-to-end delay, grouping loss rate, shake, and cost and so on. The network must satisfy this request while transmitting the corresponding data business. QoS routing is a kind of mechanism whose routing bases on the QoS request of data flow and available network resource^[4]. QoS routing is also a kind of dynamic protocol.

In the current network, many kinds of application need to offer QoS to control, for instance the videoconference and long-distance education. The network control based on QoS is the key guarantee to support the application in Ad Hoc network. It is a focus in current network research and development.

2. AD HOC NETWORK CHARACTERISTICS AND QOS TARGETS

Ad Hoc network has different characteristics compared with regular network and wireless honeycomb network^[1,5]:

Mobile node in the network is connected by wireless channel; each of them has routing functions and works as wireless router. The whole network is an autonomy system composed of such kind of wireless routers.

Ad Hoc network has a characteristic of complete distribution. It can be quickly laid without the support of basis and other key communications.

The dynamical topology is the most remarkable characteristic of Ad Hoc network. The nodes in Ad Hoc network can be moved at will and the network topology will change with nodes. In the other hand, Ad Hoc network topology is also influenced by change of characteristic of transmission and receipt of the mobile communication unit.

Ad Hoc network has such characteristics as limited link bandwidth. Variable capacity, low speed, high error bit rate, limited bandwidth resource, etc. These characteristics may lead Ad Hoc network to be in an environment of changing bandwidth and delay, and unreliable communication link.

The existence of unidirectional channel, emissive power of node, battery energy, geographical position, etc. may lead to the existence of unidirectional link.

Ad Hoc network node depends on battery to supply power. The electrical energy is limited in operation course.

According to the requirement of QoS definition, the network which can provide QoS guarantee must offer satisfied service in transmission delay, grouping loss rate, available bandwidth, shake, cost, and so on. But because of Ad Hoc network's characteristics, the routing algorithm that leads to the existence of traditional regular network and support QoS cannot be directly applied to Ad Hoc network.

To practice QoS control in Ad Hoc network, we will have to consider such problems as how to dispose network resource dynamically, how to raise data transmission efficiency, how to ensure multimedia transmit service quality, etc. Because Ad Hoc network is multihop wireless connected, it needs to ensure the QoS in the whole multihop wireless routing, so it will face the following problems: Dynamic change of topology brings some difficulties to QoS control, that request QoS routing protocol to build up a new routing quickly to meet the change. Visiting on the sharing channel which is based on competition mechanism, we will face such problems as "conceal the terminal", "expose the terminal" and "invade the terminal", etc. The solution of such problems as limited network resource, limited wireless channel bandwidth, unstable and easy to be interfered channel quality, may lead to the fact that the network is congested further. The mobile node is not suitable for transmitting a large amount of data flows as keynote is in high-speed network. When all the transmitted data flows have QoS demands, the bandwidth of wireless channel will be saturated quickly. The node needs to maintain and

* Supported by the National Natural Science Foundation of China under Grant Nos:69972036,90104002.

upgrade a large amount of link-way status information. It is a test for the limited battery and memory. The unidirectional link may emerge in any time. This will bring about more difficulties to the maintaining and updating of routing.

3. QOS ROUTING PROTOCOL BASED ON AD HOC NETWORK

Ad Hoc network is a kind of special network, according to its unique, people divide current routing protocol into two categories^[7]: table-driven selected routing protocol and according to need selected protocol. In table-driven selected routing protocol, each node keeps a newest routing form that can be relatively stable to get to other nodes. It can change network topology through broadcasting updating information in the network. According to need selected protocol sets up the routing only when the mobile host node has requests. It keeps routing only in the course of communication and removes routing after the communication has finished.

QoS routing protocol is used to search the path that satisfies QoS request. Because Ad Hoc network has characteristics of dynamical topology and limited bandwidth, it is very difficult to make it implement QoS routing. The general method is through increasing QoS parameter's restraint to the routing, deciding the transmission path according to the available resource of the network, this way the control of QoS is realized. We will introduce several kinds of topical QoS routing protocols on the basis of Ad Hoc network as follows:

3.1 STARA

STARA (system and traffic dependent adaptive routing algorithm)^[8] protocol uses the shortest path law to calculate path. It also uses average delay to measure "the shortest" routing. So QoS can support it. When STARA carries on route grouping, factors such as wireless link capacity, queuing delay and so on are considered. Each node i uses improved end-to-end confirmation protocol to calculate average delay D for each pair of node (i, d) . The calculation method is showed in , $\lambda \in [0, 1]$ is a forgotten factor, it is used to adjust the weight relation of the historical length of delay and the current length of delay ; $k \in N$, N shows all the neighboring nodes that are included in each hop of node i . According to , we then can allocate the flow to different near by nodes, So all available paths have the same delay.

$$D_{ik}^d(t) = (1 - \lambda) \sum_{l=0}^{\infty} \lambda^l \cdot D_{ik}^d(t - l) \quad (1)$$

$$P_{ik}^d(t) = P_{ik}^d(t - 1) + \alpha(t) \cdot (D_i^d(t) - D_{ik}^d(t)) \quad (2)$$

Protocol analysis:

STARA is a kind of table-driven end-to-end protocol. It searches path through finding the path that has the smallest average delay in the routing selection. QoS can support the protocol.

Because STARA is a kind of initiative routing protocol, all the nodes have to calculate average delay periodically and update routing table. Therefore the protocol needs to be supported by certain storage capacity, and the battery energy consumption is relatively big.

3.2 LS-QoS

LS-QoS (Link State base QoS routing)^[9] protocol defines

two kinds of quotas: average wrong grouping rate and living time.

The signal quality of the link from node i to node j can be described by average wrong grouping rate $P_{e[i,j]}^k \cdot P_{e[i,j]}^k$ is a weighted average, it can be dynamically adjusted according to the newest sampling, and its corresponding weighting formula is showed as in

$$P_{e[i,j]}^k = (\delta \times P_{e[i,j]}^{k-1}) + ((1 - \delta) \times New.P_{e[i,j]}^k) \quad (3)$$

In , δ is a weighting factor ($1 > \delta \geq 0$). It decides the value weight relation of the average wrong grouping rate's historical series estimates and new sampling average wrong grouping rate. In other words, δ is the sensitive degree of the average wrong grouping rate to the signal receivment quality.

Node j is the adjacent to node i . In sampling time k , $LT_{[i,j]}^K$ is the living time of the link from node i to node j , and Max_Living_Time represents maximum living time ($Max_Living_Time \geq LT_{[i,j]}^K \geq 0$). To some wireless link, the longer living time it has, the much higher existence possibility it has, and then its "confidence level" is relative higher.

Protocol analysis:

LS-QoS protocol has fully used the characteristics of Ad Hoc network wireless link broadcasting. It uses average wrong grouping rate of the link and living time as link weight value to select routing, thus achieves the goal of QoS control.

Through determining average wrong grouping rate and living time, LS-QoS protocol guarantees stability and reliability of the routing that it searches.

The protocol needs real-time collecting and compute average wrong grouping rate and link living time. At the same time, the protocol needs periodically broadcast link state database. It raises the electrical energy consumption and the routing expenses, thus aggravating the node load.

LS-QoS protocol needs to save the table of wrong grouping rate and link living time. It will take certain storage capacity for this reason.

3.3 QRME

QRME (QoS Routing Based on Maximum Expiration Time)^[6] protocol uses the idea of mobile forecast. Through GPS support, it can obtain mobile nodes' coordinates, velocity of movement and direction, therefore the protocol can forecast the connection time between two nodes through :

$$LET = \frac{-(ab + cd) + \sqrt{(a^2 + c^2)r^2 - (ad - bc)^2}}{a^2 + c^2} \quad (4)$$

$$a = v_i \cos \theta_i - v_j \cos \theta_j$$

$$b = x_i - x_j$$

$$c = v_i \sin \theta_i - v_j \sin \theta_j$$

$$d = y_i - y_j$$

In the formula, LET is the minimum time for the two nodes to hold link. v_i, v_j are the average moving speed of the nodes, θ_i, θ_j are the node host's moving directions. (x_i, y_i) and (x_j, y_j) are the coordinates of node i and node j . R is the node

host's valid transmitting range. In this formula, if two nodes have already held link and its direction is consistent with its speed, **LET** is an infinite number, it means the two nodes can hold link continuously. If **LET** is a negative number, the two nodes cannot be connected.

Through mobile forecast, QRME protocol establishes and updates the data transmission path according to the requirement of source node. It realizes information transmission, which satisfies the QoS condition between source nodes and the target node host computer.

Protocol analysis:

Through mobile forecast, QRME protocol efficiently reduces the routing establishing time and enhances the success ratio of seeking path.

In the process of routing establishment, flooding information package has the information of path change, so, the QRME protocol can precisely forecast the state of transmission path.

QRME protocol achieves the maximum maintenance time of transmission path.

The fact that nodes need flooding information package leads to higher expenses and even causes a broadcasting storm.

QRME protocol needs the support of GPS and other hardware, and the nodes have to exude flooding information packages in the same time. All of these will consume certain battery energy.

3.4 ABGR

In the view of bandwidth loss caused by the blindness proliferation of Ad Hoc network, ABGR (advanced Bandwidth Guaranteed Routing)^[11-12] protocol uses many kinds of mechanism to limit expiration, fully considers the problem of limited node electrical energy, and attempts to lengthen the integrity of the network in the routing choosing process. The protocol also attempts to avoid the network partition caused by the earlier exhausting of electrical energy of some nodes.

ABGR carries out limited proliferation according to the physical location of the target node the moving speed and the electrical energy residual of the adjacent node. Its proliferation condition is showed in

$$C_{s \rightarrow t}^x = \{n_j \mid D_t \wedge (B(x, n_j) \geq B_{threshold}) \wedge (V_{nj} \leq V_{threshold}) \wedge (C_{nj} \geq C_{threshold}), n_j \in N_s\} \quad (5)$$

In , x represents the intermediate node in the routing choosing process from source node to target node. D_t represents the direction point to target node. $B(x, n_j)$ represents the idle bandwidth in $x \rightarrow n_j$ link. V_{nj} represents the traveling velocity of node n_j . C_{nj} represents the remainder of power source of node n_j .

In the routing request process, Source node s only sends survey information to the near node that satisfies the condition $C_{s \rightarrow t}^x$, thus the Qos request that is based on bandwidth and limited by electrical energy can be guaranteed.

Protocol analysis.

Through determining the velocity of node, ABGR

protocol can efficiently limit the proliferation range of the detecting frame. And enhance the stability of the selected path.

Through defining the stop token and the limited diffusion condition that form target node to source node, ABGR protocol can efficiently control the routing establishment expenses.

ABGR protocol takes the remainder of electrical energy as QoS' condition, in a certain degree, it limits the problem of network topologic caused by changes of power.

Each node has to periodically exchange information of transmission velocity and remainder of electrical energy periodically. It needs to consume certain electrical energy and network bandwidth.

The system needs to be supported by hardware such as GPS, and so on.

3.5 LBRM

LBRM (Local broadcast touting message)^[14] protocol is designed and improved on the base of TBP algorithm^[13]. Because of Ad Hoc network's characteristics of delay and limited bandwidth, LBRM protocol uses the definition of persistent effect link to avoid the dynamical change problem of network topology which caused by the movement of the node.

To any newly built link (x, y) , we suppose it a transient link (probably only a link existing in a short time), and define its cost as $C'(x, y) = mC(x, y)$. If the link is not separated after certain period of time, then $m = m - 1$. If the link is separated after an $m - 1$ period of time, we think this link is a persistent effect link. Its expense is $C'(x, y) = C(x, y)$. According to the definition of persistent effect link, shows the delaying and bandwidth limit request of the QoS routing problem as follows:

$$\begin{aligned} \text{Delay}_{T(s,v)} &\leq D_{\max} \\ \text{Width}_{T(s,v)} &\geq W_{\min} \end{aligned} \quad (6)$$

In (6), $T(s,v)$ represents path from source node s to target node v . **Delay** and **Width** respectively represent time delay and bandwidth in this path. D_{\max} and W_{\min} respectively represent the upper limit delay and the minimum bandwidth in real time.

Protocol analysis

Through defining persistence effect link, LBRM protocol turned the dynamical variable Ad Hoc network into a processing model which is similar to the fixed network. So the problem of network's non-precise state has been solved ingeniously.

The node only maintains the state message of local link (delay, bandwidth, cost and so on), it fully uses the partly broadcasting characteristic of Ad Hoc network and reduces the complex degree of algorithm.

Through the comparison of forerunner node and successor node. The protocol can conveniently implement the link separation examination and repairment, and adapt to the topological dynamical change situation of network.

Because LBRM protocol is a definition based on the persistent effect link and when the network topology changes quickly, it is very difficult to satisfy the control requirement of QoS besides there exist a problem of how to definite the value of m .

The protocol is a generative process that satisfies the

QoS broadcasting tree. It needs to consume certain energy and take some storage space.

4. COMPARATIVE ANALYSES

An ideal routing protocol based on Ad Hoc network QoS should fully consider the defects of the Ad hoc network, such as the group nature of the network, dynamical variable

topology, limited wireless transmission bandwidth, the existence of unidirectional link, the distributional control, the short living time the energy of mobile equipment host computer, the memory size and so on^[15]. At the same time it should fully consider the requirements of Ad Hoc network's multicasting application as well. Table 1 compares and analyzes the QoS protocols in this article in the aspects of distributional operation, unidirectional link support and so on.

Table 1 Comparing QoS routing protocols of Ad Hoc network

protocol	initiative on_demand	Distributed	Unidirectional link	QoS parameters	Multicast	Support hardware
STARA	initiative	Y	Y	Average shortest_delay	N	N
LS-QOS	on_demand	Y	Y	Error packet rate , the existence time of unidirectional link	N	N
QRME	on_demand	Y	N	The Shortest time of node-linked	Y	GPS
ABGR	on_demand	Y	N	Battery remain、bandwidth	N	GPS
LBRM	on_demand	Y	N	bandwidth、delay、cost	Y	N

Distributed operation Ad Hoc network has important applications in the aspects of natural disaster rescue, military communication and so on. For these reasons, Ad Hoc network is expected to have very high robustness. As central routing protocol is very difficult to adapt this kind of requirement, all Ad Hoc networks use the distributional operation model.

The unidirectional link support An Ad Hoc network is a wireless link communication network based on each mobile host computer's battery energy, the emissive power as well as different geographical position, the possibility of unidirectional link's existence is big. The route's unidirectional characteristic brings about new difficulties to the routing protocol implication.

QoS control parameter In QoS routing, the protocol chooses different parameters according to actual conditions. In Ad Hoc network, QoS parameters include the parameter control of the node itself, such as computation ability of node CPU, memory size, and remainder of battery energy and so on. The QoS parameters also include the link parameter such as link bandwidth, delay and so on. The QoS parameter selection is only determined by the specific service and the network resource condition. It is impossible to satisfy many QoS parameters at the same time.

Multicasting function support It has a very important significance to support multicasting in Ad Hoc network. Ad Hoc network users generally are a group joint operation community. Its main application bodies are one to many or many to many multicasting communications. In multicasting communication, the protocol can efficiently use the bandwidth and save the battery energy of the node. It is an important Ad Hoc network protocol research on how to add multicasting support in QoS routing protocol.

5. CONCLUSION

This article introduced the routing protocols which support QoS control in Ad Hoc network. Detailed analyses have

been carried out to characteristics of each protocol. Finally, this article concentrates on analyses and comparisons to aspects of multicasting function, unidirectional link, etc. Since Ad Hoc network has characteristics of variable network topology, unidirectional link, and limited wireless transmission on bandwidth and so on, new problems of the realization of the QoS routing have occurred, ways of designing the protocols to realize an ideal Ad Hoc routing to support QoS still need move further research.

6. REFERENCES

- [1]. Corson S, Macker J, "Mobile Ad hoc networking (MANET): routing protocol performance issues and evaluation considerations", RFC 2501, 1999.
- [2]. "IETF Mobile Ad Hoc Networks Working Group Charter", <http://www.ietf.org/html.charters/manetcharter.html>.
- [3]. Jubin J, Tornow J, "The DARPA Packet Radio Network Protocols", Proceedings of IEEE, Vol. 75, No. 1, 1987, pp: 21-32.
- [4]. Crawley, E., Nair, R., Rajagopalan, B. *et al*, "A framework for QoS-based routing in the Internet", RFC 2386, 1998.
- [5]. Corson M S, Macker J P, "Internet-based Mobile Ad Hoc Networking. IEEE Internet Computing", No. 07/08, 1999, pp: 63-70.
- [6]. DENG Shuguang, WANG Jianxin *et al*, "A QoS Routing Based on the Steadiest Path in Mobile Ad Hoc Networks", Computer Engineering, Vol 28, No. 9, 2002, pp: 45-47. (in Chinese)
- [7]. Royer E M, "IEEE Personal Communication", Vol 4, No. 2, 1999, pp: 46
- [8]. P. Gupta, P. R. Kumar, "A system and traffic dependent adaptive routing algorithm for ad hoc networks", The 36th Conference on Decision and Control. San Diego, California, Dec 1997, pp: 2375-2380
- [9]. YING Chun, SHI Meilin, "QoS Routing in Ad Hoc Networks", Chinese Journal of Computer, Vol. 24, No. 10, 2001, pp: 1026-1033. (in Chinese)
- [10]. Lee S J, Su W, "Ad Hoc Wireless Multicast with

- Mobility Prediction.Proc”, IEEE ICCCN/ 99 , Boston, 1999, pp:4-9
- [11]. Toh C K, “Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks”, IEEE Communication Magazine. No.06, 2001, pp: 138-147
- [12]. WU Xiaobing,HUANG Chuanhe etc. “A Bandwidth Guaranteed Routing Algorithm in Moblie Ad Hoc Networks”, Computer Engineering And Applications, No.12,2003,pp : 177-180.(in Chinese)
- [13]. Shigang Chen, Klara N, “Distributed quality-of-service routing in Ad-hoc networks”, IEEE Journal on Selected Areas in Communications, Vol.17, No.8 1999, pp: 1488-1505.
- [14]. SHI Jian,ZOU Ling, “A QoS Based Distributed Multicast Routing Algorithm in Ad Hoc Networks”, Journal of China Institute of Communications,Vol.24,No.6, 2003,pp: 60-68.(in Chinese)
- [15]. Malkin G,“RIP Version 2 Carrying Additional Information”, RFC1723,1994. <ftp://ds.internic.net/rfc/rfc1723.txt>

An Improved Multicast Routing Algorithm with Delay-constrained Based on Genetic Algorithm

Wei Fang Wenbo Xu

School of Information Technology, Southern Yangtze University

Wuxi Jiangsu 214036, China

E-mail: fw_fangwei@163.com Tel: (0)13812532566

ABSTRACT

Computing the delay-constrained least-cost multicast routing tree is an NP-complete problem. In this paper, we propose an improved multicast routing algorithm based on genetic algorithms (GA). In the proposed algorithm, we use a novel algorithm of the selection of spare path as the spare paths of GA. Also we proposed a method to deal with dynamic joining and leaving of the nodes, the method is distributed. In the simulation, we use a novel algorithm to generate a random networks model. The execution time of the proposed algorithm is shorter than some of the similar algorithms.

Keywords: multicast routing; genetic algorithm; delay-constrained; distributed

1. INTRODUCTION

Multicast is a kind of group communications which require simultaneous transmission of messages from a source to a group of destinations. The real-time multicast refers to a multicast in which sending the same data from a source to a group of destinations in a computer or communication network with a specified time delay. Recently, advances in media and switch technologies have resulted in a new generation of wide area networks. These networks are expected to support a wide range of communication-intensive real-time multimedia applications like digital audio and video. The deployment of high-speed networks opens a new dimension of research, providing quality of service (Qos) for multimedia application. But it is a technically a challenging and complicated problem to deliver multimedia information in a timely, smooth, synchronized manner over a shared network environment, especially one that was originally designed for best-effort traffic such as Internet.

In the past, most of the applications were unicast in nature and none of them had any Qos requirements. However, with emerging distributed real-time multimedia applications, the situation is completely different now. These applications will involve multiple users, with their own different Qos requirements. Support point to multi-point connections for multimedia applications requires the development of efficient multicast routing algorithms.

Routing is an important part of connection setup. It is to select a route from the source to destinations in which the connection will be established. The multicast routing is to find a routing tree, which rooted from the source and contains all the destinations. The network cost of a multicast routing tree can be measured by the sum of the cost of all the links in the tree. One requirement of multicast routing is to minimize this network cost. A tree with minimal overall cost called a Steiner tree [1]. The Steiner tree problem [2] tries to find the least-cost

tree, the tree covering a group of destinations with the minimum total cost over all the links. Finding such a tree in a network is a problem of NP-complete [3]. Most of the proposed algorithms for Steiner tree are heuristic. Some of the well-known Steiner tree heuristics are the RS heuristic [4], the KMB heuristic [5]. Several algorithms based on neural networks [6] and genetic algorithms [7] have been also proposed for solving this problem. In addition to the minimal network cost, real-time multicast routing has a bounded delay requirement. Many real-time applications, such as interactive multimedia applications, often have time constraint, which requires the communication to be within the pre-specified delay bound. Therefore, multicast routing for real-time applications is to find a routing tree whose network cost is minimal and the delay from source to any destination does not exceed a pre-specified bound. This is called constrained Steiner tree problem [8]. Early studies have considered the delay and cost optimization objectives separately. Dijkstra's shortest-path algorithm can be used to compute a delay-preserving multicast tree in polynomial time. The minimum spanning tree-based greedy heuristic algorithm proposed in [4] can compute in polynomial time and the cost is at most twice that of an SMT (Steiner minimum tree). The best-known polynomial time approximation algorithm to this algorithm is proposed in [9], which computes a multicast tree whose cost is about 1.55 times the cost of an SMT. Bharath-Kumar and Jaffe [10] discussed minimizing both cost and delay in a multicast tree, assuming that the cost and delay functions are identical. Khuller proposed a best-possible algorithm to balance a minimum spanning tree and a shortest-path tree (SPT) [11]. This idea has been used by Lin and Xue [12], in the design of performance-driven rectilinear SMTs. Parsa et. [13] Proposed an algorithm which refines an SPT to a low-cost delay-constrained multicast tree by iteratively replacing the most expensive super-edge by a less expensive path, without violating the delay constraints. Jia [14] and Jia et [15] proposed efficient admission-control methods based on the balance of SMTs and SPTs. Recently a lot of delay-constrained least-cost multicast routing heuristics such as the KPP heuristic [16], the BSMA heuristic [13] and so on. But heuristic algorithms for Qos multicast routing are usually very slow, methods based on computational intelligence such as neural networks and genetic algorithms may be more suitable.

Genetic algorithm (GA) is a new optimize algorithm that proposed in recent years. It has the characteristics of parallel search, population optimization, and so on. GA has widely used to solve the questions, which is NP-hard. Xiang [17] et.al have proposed a GA-based algorithm for Qos routing in general case. This algorithm adopts an N*N one-dimension binary encoding scheme, where N represents the number of nodes in the graph. However, in this encoding scheme, the transformation back and forth between genotype and phenotype space is very complicated, especially for large networks. Ravikumar et [18] have proposed a GA-based

algorithm with novel interesting approaches for crossover and mutation operators for the delay-constrained least-cost multicast routing problem. However, they have not defined their scheme for encoding and decoding of individuals. But their algorithms may lead to premature convergence. Zhang [19] have proposed an effective orthogonal GA for delay-constrained least-cost multicast routing algorithm. This algorithm also assumes the delay constraints for all destinations are identical. Wu et [20] have proposed a GA-based algorithm for multiple QoS constraints multicast routing problem in general case. However, their genotype representation dose not necessarily represents a tree. It is necessary to construct and store a very large amount of possible routes for each pairs of nodes in the graph using the K-shortest path algorithm.

In this paper, we propose an improved multicast routing algorithm base on genetic algorithms. A novel function is used to initial colony. Some novel heuristic algorithms are also proposed for mutation, crossover, and creation of random individuals. We also evaluate the performance and efficiency of the proposed GA-based algorithm in comparison with some other similarity algorithms. After constructed the multicast tree, we should consider that in the real networks, such as distance learning and teleconferencing, multicast participants are free to leave or join a multicast tree dynamically. So we should design a method to ensure that any change of multicast participants will not affect the traffic of the current connection and the tree cost remains sub-optimal after change.

This paper is organized as follows. In section 2, we describe and formulate the problem. In section 3, we describe the proposed algorithms. Section 4 proposes a method to control the nodes that dynamically leave or join the multicast tree. In section 5, gives the comparison of the proposed algorithm with some other similarity algorithms. Section 6 concludes this study.

2. NETWORK MODEL AND PROBLEM DEFINITION

In this section, we formally define the problems that will be studied in later sections. The communication network is modeled using an edge-weighted directed, connected graph $G=(V, E, c, d)$, where V is the set of n vertices (network nodes) and E is the set of l edges. Here $n=|V|$ be the number of network nodes, $l=|E|$ be the number of network link. And c, d is two non-negative real value functions which associated with each link $e (e \in E)$: Delay $D(e)$ and cost $C(e)$. The link delay $D(e)$ is the delay a data packet experiences on the corresponding link, and the link cost $C(e)$ is a measure of the utilization of the corresponding link's resources. Links are asymmetrical, namely the cost and the delay for the link $e=(i, j)$ and the link $e'=(j, i)$ may not be the same.

Given a source node $s (s \in V)$, a set of destination nodes $M (M \subseteq V-s)$, a tree $T (T \subseteq G)$ rooted at s and spanning all of the nodes in M (with all leaf nodes $\subseteq M$) is called a multicast tree. Let $m=|M|$ be the number of multicast destinations nodes. We refer to M as the destination group and $\{s\} \cup M$ the multicast group. In addition, $T(s, M)$ may contain relay nodes (Steiner nodes), that is, the nodes in the multicast tree, but not in the multicast group.

The total cost (C_T) of the tree $T(s, M)$ is defined as Eq.(1):

$$C_T = \sum_{e \in T} C(e) \quad (1)$$

The total delay of the path $P_T(s, d)$ is simply the sum of the delay of all links along $P_T(s, d)$ as Eq.(2):

$$D(P_T(s, d)) = \sum_{e \in P(s, d)} D(e) \quad (2)$$

For a network $G=(V, E)$, given a link delay function $D: E \rightarrow \mathbb{R}^+$, a link cost function $C: E \rightarrow \mathbb{R}^+$, a source node $s (s \in V)$, a set of destination nodes $(S \subseteq V-s)$, and a delay constraint Δ , find a multicast tree $T (T \subseteq G)$ rooted at s and spanning all of the nodes in S such that C_T is minimized and

$$\sum_{e \in P(s, d)} D(e) < \Delta \quad (3)$$

The choice of Δ depends on the applications .

3. THE PROPOSED GA-BASED ALGORITHM

In general, GA can be divided into five steps as shown in Fig.1.

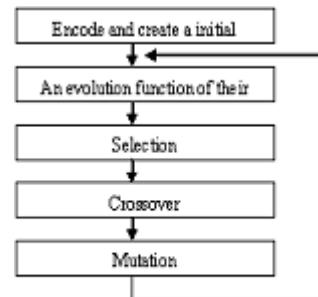


Fig.1 Five steps of GA

3.1 Selection of Spare Path

Some of the GA algorithms use Dijkstra's k-shortest-path algorithm to find the paths that satisfy Eq.(3). In our paper, we use another approach [21] to find those paths, which satisfy Eq.(4). The approach is as follows:

$$\begin{cases} f(v, w) = \frac{c(v, w)}{(\Delta - P(v) - D(v, w))^k} \\ P(v, w) + D(v, w) < \Delta, \text{ otherwise, } f(v, w) = \infty \end{cases} \quad (4)$$

$K \geq 0$ and $P(v)$ is the delay from source s to destination v , $c(v, w)$ and $D(v, w)$ is the cost and the delay of link (v, w) . The function is synthesized and extended by the function that proposed in [22]. When k is relative great, the proportion of denominator is bigger. If k is bigger enough (usually above 20), we can get almost the same paths with the paths that get by Dijkstra's algorithm under the same delay-constrained. When k is bigger, the operation time is increased and may be

$$\begin{cases} f(v, w) = \log(c(v, w)) - k * \log(\Delta - P(v) - D(v, w)) \\ P(v) + D(v, w) < \Delta \\ \text{otherwise, } f(v, w) = \infty \end{cases} \quad (5)$$

overflow. We can change Eq.(4) into Eq.(5):

We can get a set of paths as the spare paths of GA. Then the search path of GA is restricted in the set that we have got.

3.2 Encoding

Let N_m as the number of the paths that satisfy delay-constrained from source s to destinations $m \in M$, then we can correspond these paths with $R_m = \{1, 2 \dots N_m\}$. Since the search path is restricted in the spare path, the individual of GA can denote by $|M|$ -dimension non-negative integer and the coding is $\{a_1, a_2 \dots a_{|M|}\}$, $a_m \in R_m$. Then select a number from each R_m randomly and compose a multicast tree's coding as a chromosome of initial population.

3.3 Fitness Function

The fitness function in our study is an improved version of the scheme proposed in [23]. We define the fitness function for each individual, the tree $Tree(s, M)$, using the penalty technique, as Eq(6):

$$\phi(z) = \begin{cases} 1, z \leq 0 \\ r, z > 0 \end{cases} \quad (6)$$

$$F(T(s, M)) = \frac{\alpha}{\sum_{e \in T(s, M)} C(e)} \prod \phi(D(P(s, d)) - \Delta_d),$$

where α is a positive real coefficient, $\phi(z)$ is penalty function and γ is the degree of penalty. If there is repeat link that linked to different destinations, then the cost of the link only computer once.

3.4 Selection

The selection process used here is based on spinning the roulette wheel of fitness value. The probability of fitness value is adjusted as follows:

$$f' = \alpha * f + \beta, \quad \alpha = \frac{f_{avg}}{f_{avg} - f_{min}}, \quad \beta = \frac{-f_{min} f_{avg}}{f_{avg} - f_{min}}$$

f_{avg}, f_{min} is the mean and minimum of present population's fitness value.

3.5 Crossover

In our paper, we first use one pointer crossover operator, with a fixed probability P_c . The constructed offspring do not necessary represent Steiner trees. Then use the effective and fast check and recovery algorithm proposed in [19] to connect the separate sub-trees in the offspring and also connecting the absent nodes of multicast group to the final tree. [24]

3.6 Mutation

We use the mutation algorithm proposed in [25] which is one point mutation and self-adjusting mutation probability, $P_m = c'(f_{max} - f)/(f_{max} - f_{min})$, f_{max}, f_{min} are the maximum and minimum adjusting value of the current population is a constant.

4. DYNAMIC JOINING AND LEAVING

Based on the multicast tree that constructed by our GA algorithm, we set up a table which named Mutable for each node of the tree, Mutable include three items, Mutable. cost, Mutable. delay, Mutable. tag and they respectively denote the cost of the node to the source, the delay to the source and the tag if it has in the multicast tree.

4.1 Dynamic Joining

When there are several nodes v_1, v_2, \dots, v_s , which take parameters D_v and Δ request to join the multicast tree, the process can be described as follows [27].

Step1: randomly select a node named as n_1 , which is in the multicast tree, at the same time get the correspond info of the node from table Mutable

Step2: compare the path from n_1 to the nodes that request to join, and then select a node named as v_1 that has the minimum path connect to n_1 .

Step3: compare the path form v_1 to the nodes in the multicast tree, and then select a node named as n_1 that has the minimum path connect to v_1 .

Step4: $D = D_v + D_{v_1}$, D_{v_1} is the delay of the node v_1 , it can be attained from the table Mutable; if $D < \Delta$, add the node into the multicast tree, and append the cost, delay, tag of the node into the table Mutable; if $D > \Delta$, then the node n_1 send a message of reject to the node v_1 ; go to step1, and select the least minimum path.

4.2 Dynamic Leaving

If a destination node of the multicast tree request to leave, the question can be divided into two instances.

Leaving 1: the destination is a leaf node of the multicast tree, it just sends a message to its parent node that will terminate its connection to the leaf node, at the same time, delete the info in the Mutable of the leaf node

Leaving 2: the destination is not a leaf node; it simply changes its status from destination node to relaying node.

5. SIMULATION AND COMPARISONS

5.1 Random Network Generating Model

In many literatures, they use the algorithm of randomly generating network topologies that describe in [26]. We use the method of generating a random network that proposed in [27], the nodes are distributed randomly over a rectangular coordinate grid. Each node is placed at a location with integer coordinates. A link between two nodes u and v is added by using the probability function

$$P(u, v) = \beta M_1 / (n \log_0 \exp[d(u, v) / (L M_2 / \log n)]),$$

α, β are the parameters used to adjusting network's character. When α is increased, the ratio of long side and short side is increase; when β is increased, the degree of nodes is increased. In our simulation, $\alpha = 0.4, \beta = 0.8, u = 4.00$, the average degree of nodes is 4.

5.2 Experimental Results

A random network model is generated by the means mentioned above. Before we begin the GA operations, we have used an method (see in 3.1) to select the paths. So the GA algorithm can operate less paths in the network.

Figure 2 shows the example of the execution time of our proposed GA-based heuristic algorithm in comparison with the mentioned existing algorithms. This figure shows that our proposed algorithm can result in a smaller execution time than the mentioned existing algorithms.

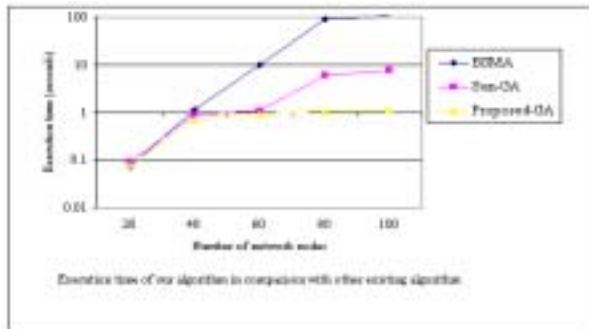


Fig.2 The execution time

Figure 3 show the percentage tree cost of BSMA, Sun GA-based heuristic algorithm in comparison with our proposed algorithm for different network sizes and different multicast group sizes. The figure shows that our proposed algorithm can result in a smaller average tree cost.

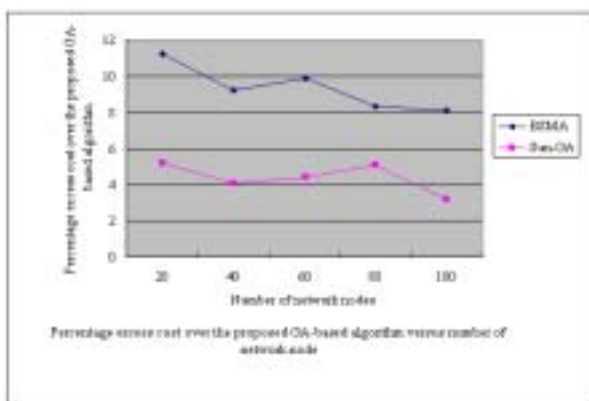


Fig.3 The percentage tree cost of BSMA

6. CONCLUSION

In this study, we have proposed a GA-base heuristic algorithm to solve the delay-constrained least-cost multicast routing problem which is known to be NP-complete. The simulation results have shown that the execution time of the proposed algorithm is shorter than some other algorithms. In this study, we have focused on the selection of spare path.

7. REFERENCES

- [1] Gibert and H.O.Pollak,"steiner minimal tree", SIAM J,Appl Math.,vol,16,1968
- [2] S.L.Hakimi, Steiner problem in graphs and its implications, Networks 1(1971) 113-133

- [3] M.R.Garey and D.S.Johnson.Computers and Intractability: A Guide to the Theory of NP-completeness, San Francisco, CA: freeman, 1979
- [4] H.Takahashi, A.Matsuyama, An approximate solution for the Steiner problem in graphs, Mathematica Japonica 22(6)(1980), 573-577
- [5] L.Kou, G.Markowsky, L. Berman, A fast algorithm for Steiner trees, Acta Informatics 15(1981) 141-145
- [6] E.Gelenbe, A.Ghanwani, V.Srinivasan, Improved neural heuristics for multicast routing, IEEE Journal of selected Area in Communication 15(2)(1997) 147-155
- [7] Y.Leung, G.Li, Z.B.Xu, A genetic algorithm for the multiple destination routing problems, IEEE Transactions on Evolutionary Computation 2 (4)(1998) 150-161
- [8] V.P.Kompella, J.C.Pasquale and G.C.Polyzos,"multicast routing for multimedia communication", IEEE/ACM Trans on Networking, Vol.1, No.3, June 1993,pp286-292
- [9] G.Robins and A.Zelikovsky,"Improved Steiner tree approximation in graphs ", in Proc.11th Annu.ACM-SIAM Symp.Discrete Algorithms, 2000,pp770-779
- [10] K.Bharath-Kumar and J.M.Jaffe,"routing to multiple destinations in computer networks ",IEEE Trans.commun.,vol.31,pp343-351,Mar.1983
- [11] S.Khuller, B.Raghavachari, and N.Young,"Balancing minimum spanning and shortest path trees, " in Proc.AcM/SIAM SODA, 1993,pp243-250
- [12] G.H.Lin and G.L.Xue,"Balancing Steiner minimum trees and shortest path trees in the rectilinear plane", in Proc.IEEE ISCAS,vol.VI,1999,pp117-120
- [13] M.Parsa,Q.Zhu, and J.J.Garcia-Luna-Aceves,"An iterative algorithm for delay-constrained minimum-cost multicasting",IEEE/ACM Trans.Networking ,vol 6,pp.461-474,Aug.1998
- [14] X.H.Jia,"A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," IEEE/ACM Trans,Networking,vol 6,pp.828-837,Dec.1998
- [15] X.H.Jia, Y.B.Zhang, N.Pissinou, and K.Makki,"An efficient admission control method of real-time multicast connections in wide area networks," in Proc.IEEE ICCCN, 1998, pp.865-872.
- [16] V.P.Kompella,J.C.Pasquale,G.C.Polyzos,Multicast routing for multimedia communication,IEEE/ACM Transactions on Networking 6(4)(1993) 286-292
- [17] F.Xiang,L.Junzhou,W.jieyi,G.Guanqun,Qos routing based on genetic algorithm,Computer Communications 22(1999) 1394-1399
- [18] C.P.Ravikumar,R.Bajpai,Source-based delay-bounded multicasting in multimedia networks, computer communications 21(1998),126-132
- [19] Q.Zhang,Y.W.Lenug,An orthogonal genetic algorithm for multimedia multicast routing ,IEEE Transactions on Evolutionary Computation 3(1)(1999) 53-62
- [20] J.J.Wu,R.H.Hwang,H.I.Lu,Multicast routing with Qos constraints in ATM networks, Information Sciences 124(2000) 29-132
- [21] Y.Huang, An effective delay-constrained multicast routing algorithm, computer and information technology, 2002,vol.6
- [22] Kompella V P, Pasquale J C,Polyzos G C, Multicast routing for multimedia communication, IEEE/ACM Trans Networking,1993,1(3):286-292
- [23] Z.Wang, B.Shi, E. Zhao, Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm,Computer communication 3(1)(1999) 53-62

- [24] A.T.Hlaghighat .K.Faez, M.Deaghan ,GA-based heuristic algorithms for QoS based multicast routing, Knowledge-based systems 16(2003) 305-312
- [25] J.Shi,L.Zhou,The application of genetic algorithm in multicast routing ,Acta electronica sinica,May 2000,Vol.28
- [26] Waxman,B.M,Routing of multipoint connections,IEEE Journal on Selected Areas in Communications, 1988,6(9):1617-1622
- [27] H.B.Han,J.P.Yu,W.X.Xie, Fast algorithms for multicast tree construction in wide area networks ,Journal of computer research and development,Nov.2000,Vol.3



Wei Fang is a candidate for PhD at the School of Information Technology, Southern Yangtze University. He graduated from Southern Yangtze University and received his BE in 2002. His research interests include multicast routing algorithms, Genetic Algorithm.



Wenbo Xu is a full professor and dean of School of Information Technology, Southern Yangtze University. He graduated from Tsinghua University in 1968. In 1981, he graduated from Tsinghua University and acquired Master's degree with specialty of theoretical electrotechnics. He was a

visiting scholar and a visiting professor of University of Toronto in 1987 and in 2000, respectively. He is one of the DCABES international conference founder, was the chairman of DCABES 2002. He has brought to success more than 30 scientific research programs since 1990 and published over 100 papers. His current research interests are in distributed and parallel computing, intelligent computation, quantum computation and computational finance.

The Distributed Interactive Multimedia Synchronization Model Based On the Temporal Petri net

Lu Feng , Guo Yingli

The School of Information Engineering, Wuhan University of Technology
Wuhan, Hubei, China

Email: lufengwut@163.com Tel : 02787870877

ABSTRACT

The distributed multimedia synchronization is one of the most popular contents of computer science and technology. Petri Net is fit for analyzing the synchronization mechanism of distributed multimedia. This paper introduces the distributed multimedia synchronization based on the Petri net. Then it proposes and analyzes OCPN and some temporal Petri net such as TSPN and DTPN. It also discusses the interoperable Petri net. Based on these models, we put forward ISPN (distributed interactive multimedia synchronization based on Petri net) model, and interpret this model in detail. By using the model, the multimedia synchronization in distributed settings can be described accurately and effectively.

Keywords: Distributed multimedia; Multimedia synchronization; Object composition Petri net; Interoperable Petri net; Interactive multimedia environment

1. INTRODUCTION

With the development of computer and network technology and the large requirement on information, the distributed multimedia communication system is widely applied, such as remote teaching, remote therapy and so on. In this application system, the synchronization of media information is a key technology.

In the distributed environment, the source port and the destination port of media information are always distributed in different places. In destination port, the media information represented are obtained by network communication. Obviously, distributed multimedia system based on network is more complex than undistributed system. Compared to the traditional multimedia system, distributed multimedia system has 3 important characteristics: decentralization of media, continuous exchange and real-time synchronization. The data type processed in multimedia system concludes: figure, image, text, audio and video. The composition and representation of multimedia is arranged by temporal order and special relation, so the multimedia synchronization is complex. It becomes one of the important problems in multimedia technology.^{[1][4]}

Many methods of describing temporal relation were put forward in multimedia representation. These methods can be divided in two types: The first is belonged to specification method of state transfer, including the model based on Petri net, the model of data stream graph, extended finite state automatic mechanism and so on. The second is specification method based on programmed language. Generally speaking, every technology of formalization should be extended in order to meet the requirement of real-time and

synchronization in distributed multimedia. The specification method of state conversion can represent controlling direction clearly and parts of the data be expressed concealed. Petri net is explored as a concurrent model, so it realizes the concurrency and interoperation.

As a tool of mathematics, Petri net can describe the activity of system by setting arrival graph, state equation and some other formal methods. The arrival graph of Petri net is one of the primary methods of Petri net model. The temporal logical analyzed method is an important formal tool in describing and validating concurrent characteristic.

2. THE MULTIMEDIA SYNCHRONIZATION BASED ON PETRI NET

2.1 Some Synchronization Model

Petri net is a mathematic and graphic tool to describe the activity of system. Petri Net model describes the dynamic performable semantic existing among distributed multimedia objects: it illuminates activities of order, concurrent and synchronization, and meets the requirement of multimedia representation. Because of such specialties, Petri net is fit for describing the temporal model.

Definition 1: Petri net is a three-tuple, $C_{PN} = \{T, P, F\}$, it is defined as shown below: $T = \{t_1, t_2, \dots, t_n\}$ is a set of transitions, where $n \geq 0$; $P = \{P_1, P_2, \dots, P_m\}$ is a set of places, where $m \geq 0$ and $P \cap T = \emptyset$; $F : \{T \times P\} \cup \{P \times T\}$ is a set of directed arcs.

2.1.1 Object composition Petri net

The model increases the plus work time and necessary resource on the node of Petri net. And at the same time, it keeps the instantaneous of starting transitions. Every transition is the synchronization node among objects. It is used to describe the normative Petri net model of multimedia synchronization primitively. It is formed on the base of general Petri net by increasing value of delay-time and resource. It can describe the relationship of multimedia synchronization availably.

Definition 2: OCPN is defined as $C_{OCPN} = \{T, P, F, D, R, M\}$. The definition of T, P, F are the same as definition 1. $M: P \rightarrow \mathbb{N}$, $I = \{1, 2, 3, \dots\}$ is the set of the token number owned by the node. $D: P \rightarrow \mathbb{R}$ (set of real number) is the representing time of node, $R: P \rightarrow \mathbb{R}^k$, $\{r_1, r_2, \dots, r_k\}$ is a set of recourse needed by node representing.

OCPN distribute the time of reappearance resource and output reappearance data for every place. The rules of firing are as follows:

When all input places of a transition t_i contain

unlocked tokens, transition t_i fires immediately.

After the transition t_i fires, the tokens of every input places were deleted, and a token was increased on every output place.

After a place p_i obtains a token, it keeps the available and locked state during the time of running. When this place is of no effect or excess the locked interval, it tuned to unlocked state.

OCPN is a synchronization model of coarse granularity. It can show the synchronization criterion and temporal explanation of multimedia system. Especially, it is fit for describing the synchronization relationship of multimedia object in the applications of multimedia.^[2]

2.1.2 Extended OCPN Model

Two extended OCPN will be introduced: TSPN: Time Stream Petri Net; DTPN: Dynamic Time Petri Net.

Definition 3 : $C_{TSPN}=\{T,P,F,B, M_0,IM,SYN,MA\}$, the definition of T,P is the same to definition 1, $B: P \times T$, $I_i=\{1,2,3,\dots\}$, is a set of arc from place to transition; $F: T \times P$, $I_i=\{1,2,3,\dots\}$, is a set of arc from transition to place; M_0 is the primary token; IM is the temporal mapping function; SYN is the type of synchronization transition, representing different dynamic synchronization semantic; MA is the mapping relationship between Master transitions and related Master places. But in actual network environment, it can't deal with the user interactive synchronization in multimedia

representation.

Definition 4: $C_{DTPN}=\{T,P,F,D,R,M,C,E\}$, the definition of T,P,F,D,R,M is the same to definition 2; $C=\{P \times T\}$ is the set of output arc; $E: P \rightarrow R$ is the remainder interval of place. But it doesn't solve the dithering in distributed environment. Compared to the OCPN, the firing rule increases several parts:

To the transition t_i of output arc, if other places of transition t_i are in active states, include a locked token, and at least one output interface include tokens, t_i can be executed ahead of schedule. After t_i is executed, every input places will delete a token and every output places will increase a token;

If a active place is executed ahead of schedule, the remainder interval of this place must be changed.

2.2 Enhanced Priority Petri Net

Compared to the general Petri net, enhanced Petri net introduce multi-priority and dynamic arcs. It is the expansion of Petri net. After higher priority finished, a transition will fire immediately. If there is not any other requirement of priority, the default priority is 1. So the higher priority is required, the higher priority of the arcs is needed. Based on it, the distributed multimedia synchronization model is more applied to describe the distributed environment.^{[3][5]}

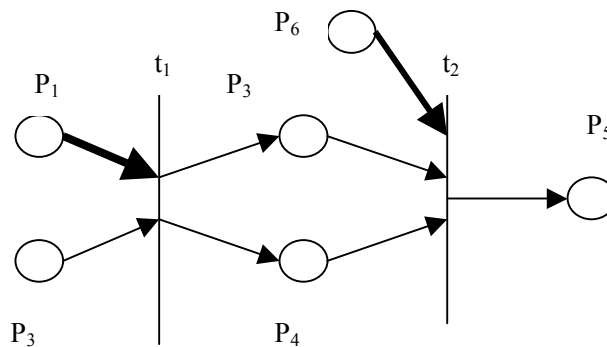


Fig 1. an instance of enhanced priority Petri net

A transition with non-priority input events only would fire when all events are complete and ready. A transition with a priority input event concurring could fire with the arrival of that priority input event, without waiting for other non-priority events. For the same priority input events concurring at a transition, we apply the "AND" rule. A place with a token and several transitions enabled from this place will fire the transition with a priority arc from this place. If there are more than one priority arc outgoing from a place enabling more than one transition, then the firing choice is non-determinate. Figure 1 is an instance of enhanced priority Petri net.

In the transition t_1 , t_1 will fire after input events arrive. In transition t_2 , the arriving of input event p_3 or p_4 will not fire the t_2 while the arriving of p_6 can. So, t_1 , t_2 can be the synchronization point of the distributed multimedia system. The rule of firing can meet the requirement of most

synchronization.

2.3 The Interoperable Petri Net (IPN)

To overcome the limitations of the TSPN and DTPN models, we introduce the IPN model to represent temporal relations. The IPN model preserves the interoperability characteristics of multimedia applications. Two or more objects, agents, or applications are interoperable if they can cooperate, regardless of their underlying technology, to accomplish a joint task. Previously, IPN is used to model the intra-stream and inter-stream synchronization requirements. Here, the IPN model is used in the formal specification and analysis of distributed multimedia synchronization agents, basing the agents' implementation on the IPN's specifications.

An IPN model, informally, consists of a processing element, ports, and port interfaces. An high level Petri net represents the processing element; ports can be input or output; and

port interfaces are the visible part of the IPN module, where data can be consumed or produced. For two (or more) IPN to interoperate, a virtual connection must be established between the source's output-port interface and the destination's input-port interface.

Before any transition can be enabled in an IPN, all variables in the arc label function must be bound to the corresponding types. According to this binding, if the guard is evaluated to true, the transition fires, thus removing tokens from its input place and adding tokens to its output place. The number of removed and added tokens is determined by the value of the corresponding arc label function. If a transition is an output port of an IPN module, its firing produces tokens that are added to its port interface. Consequently, if the interface is at its normal state, it will be promoted to the excited state. If a place is an output port, after receiving a token, the arc label function of the arc connects the port to its interface, which will then be evaluated and bound to the corresponding types. According to the binding, the port interface will be promoted into the excited state only if the guard is evaluated to true.

The notion of time is incorporated into IPN. IPN is used to model multimedia synchronization, and the synchronization deals with the temporal behavior of related streams. With temporal IPN, it is free to choose the granularity of synchronization by assigning time durations to place.

3. THE DISTRIBUTED INTERACTIVE MULTIMEDIA SYNCHRONIZATION MODEL BASED ON THE TEMPORAL PETRI NET

In a distributed environment of multilevel interconnection network and heterogeneous, the models such as OCPN can't describe the distributed multimedia synchronization perfectly because of their own deficiencies. On the basis of above models, in this paper, some deficiencies are overcome, and we put forward a distributed interactive multimedia synchronization model based on the temporal Petri net (ISPN). This model introduces the response of priority. It uses different firing of order for different input events, and integrates some characteristics of TSPN and DTPN—dynamic synchronization and interactive function. So ISPN is provided with interactivity and priority. On the controlling of the whole dynamic, the temporal synchronization of inter-media and intra-media can be described better.

3.1 The Definition of ISPN

Definition 5: $C_{ISPN} = \{T, P, B, F, D, R, M, S, X, SYN, MA, TP, TM\}$, the definitions of T, P, B, F, D, R, M are the same as definition 2; $S : P \rightarrow \{s_1, s_2, s_3, \dots\}$ is the mapping relationship of places and servers; $X = \{x_1, x_2, x_3, \dots\}$ is the set of interactive buttons, representing several buttons: reverse, skip, restart, modify speed, pause and multicasting. The multicasting applications can be implemented based on a multithreaded server/clients solution. The server is preferably situated on a multiprocessor platform to ensure quick response time to service each multicasting client. If tight synchronization is required, priority arcs can be used to enforce tight deadlines when delivering/presenting media contents to each client. In the case when flexibility is offered to multicasting clients to change viewing conditions, dynamic arcs associated with program statements can come to rescue to change the

arrangement of resources and implement corresponding schedule update; $D : P_i \rightarrow \{[a_i, n_i, b_i]\}$ is the mapping relationship of places and representing period of time. a_i is the minimum period of time, n_i is general period of time, b_i is the maximum period of time. And $0 \leq a_i \leq n_i \leq b_i$; $SYN : T \rightarrow \{And, Or, Strong-Or, Weak-And, Strong-And, Master, Or-Master, And-Master, Weak-Master, Strong-Master, X\}$ is the type of synchronization transition, describing the different dynamic synchronization semantic among several media objects while many events inputting in the transition; MA : is the mapping relationship between Master transitions and related Master arcs. $T \rightarrow A$, for any $t_j \in T$ $[SYN(t_j) \in \{master, and-master, or-master, weak-master, strong-master\}]$, there is a $A_j | MA(t_j) = a$, $A_j = \{(p_i, t_j) \in A\}$; TP : is the time mapping function, $TP(t_i) \in [0, \tau_i]$ is the maximum tolerable time that place p_i unlock after transition t_i fired, called tolerable blocking time. $TM : T \rightarrow C$ is the map of transition to set C , $C = (0, 1)$ is the symbol of transition firing, the value change to 1 from 0 while firing.

3.2 The State of ISPN

The state of ISPN S is a set of three-tuple (P, T, X) :

P is the set of all places containing tokens;

T is the dynamic valid period of time list of places in P . The number of its items corresponds to the number of places marked in the set P . When users execute the operation of "pause", T is used to record the remaining time of the operation, viz. the time of the next re-operation, in order to use in the next restarting.

X is the list of valid button, used to record the current interactive operation.

After the transition occurred, a new state S' will be formed. Supposing $t_n \in T$ is the transition of state S enabled in the relative time τ . After transition t_n fired, the new state $S' = (P', T', X')$:

P' is still the set of all places containing tokens. In P , output places of transition t_n is added, at the same time, the input places of transition t_n is deleted.

T' is the dynamic valid period of time list of places in P . T' is deduced by the follows: from the original dynamic valid list T of period of time, dynamic valid period of time of transitional input places were deleted. Supposing $[a_i, n_i, b_i]$ is the valid representing intervals of p_i in the original state S , after the transition t_n fired, p_i still contains token. In the new state S' , the valid representing interval is: $[\max(0, a_i - \tau), n_i - \tau, \max(0, b_i - \tau)]$, τ is the relative time after transition t_n fired.

X' is still the list of valid button. If $t_n \in T$ and $M'(p) > 0$, the value is 1; in other cases, the value is null. (p is the inputting place of transition t)

3.3 Transitional Interval of ISPN

Supposing P_i is the input places of transition t_n , the valid interval is $[a_i, n_i, b_i]$, the relative temporal fired range of t_n is $\tau = [\min_n, \max_n]$. The fired temporal range of synchronization transitional type can be calculated.

$SYN(t_n) = And$, $\tau = [\max(a_i), \max(\min(b_i), \max(a_i))]$; In the

all input places, after the temporal semantic is completed, the transition fires;

$\text{SYN}(t_n)=\text{Weak-And}$, $\tau=[\max(a_i), \max(b_i)]$; In the all input places, when the semantic of the last place is completed, the transition fires;

$\text{SYN}(t_n)=\text{Strong-And}$, $\tau=[\max(a_i), \min(b_i)]$; In the all input places, at least one place of temporal semantic doesn't delay, the transition fires;

$\text{SYN}(t_n)=\text{Or}$, $\tau=[\min(a_i), \max(b_i)]$; If any temporal semantic is completed, the transition fires;

$\text{SYN}(t_n)=\text{Strong-Or}$, $\tau=[\min(a_i), \min(b_i)]$; If the first temporal semantic is completed, the transition fires;

$\text{SYN}(t_n)=\text{Master}$, $\tau=[a_i, b_i]$; After the temporal semantic in the place of Master is completed, the transition fires;

$\text{SYN}(t_n)=\text{Or-Master}$, $\tau=[\min(a_i), b_i]$; In the period of the semantic completed in the place of Master, if the temporal semantic of any places is completed, the transition fires;

$\text{SYN}(t_n)=\text{And-Master}$, $\tau=[\max(a_i), b_i]$; In the period of the temporal semantic completed in the place of Master, after the semantic of all the places is completed, the transition fires;

$\text{SYN}(t_n)=\text{Weak-Master}$, $\tau=[a_i, \max(b_i)]$; In the period of the temporal semantic completed in the place of Master,

after the semantic of the last place is completed, the transition fires;

$\text{SYN}(t_n)=\text{Strong-Master}$, $\tau=[a_i, \min(b_i)]$; In a Master place or at least one non-Master place, after the temporal semantic is completed, the transition fires;

$\text{SYN}(t_n)=X$, $\tau=[0, \tau(\min(\max(a_i)), \max(a_i), \min(a_i), \min(b_i), a_i)]$ (τ is any item of the set); If at least one place concludes token, the transition fires after any button event arriving.

4. THE INSTANCE OF ISPN APPLICATION

Here is an instance of multimedia synchronization scene. ISPN is used to describe this scene. I_i is the information of picture and text, A_i is the information of audio, V_i is the information of video. The dynamic transitional process of the model represents the controlling process of the multimedia synchronization. The figure of instance is shown below:

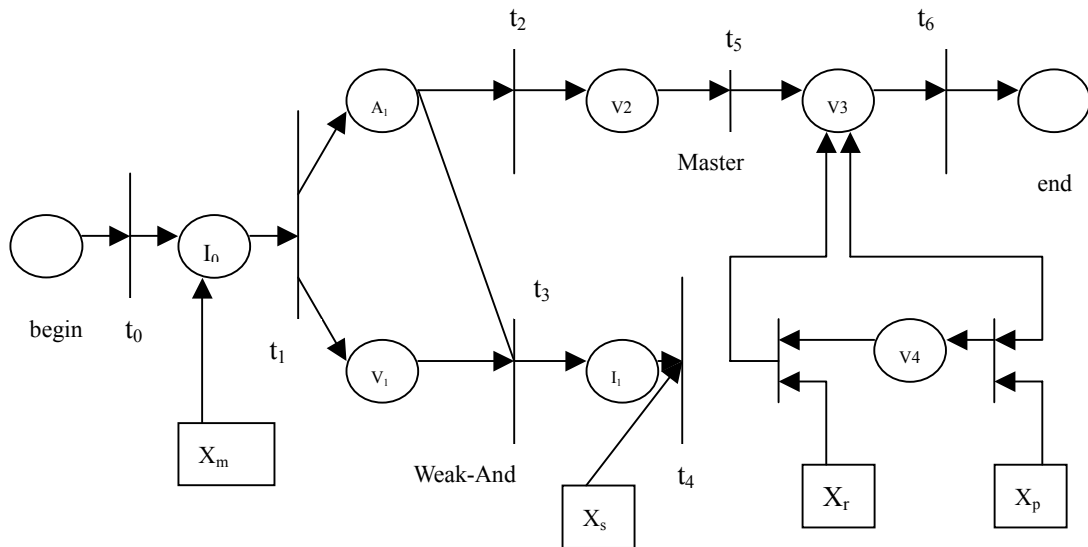


Fig 2. Instance of ISPN

The original state is fired at t_0 . $S_0=(P_0, T_0, X_0)$, $P_0=(\text{begin}(1))$, $T_0=[0,0,0]$, $X_0=0$;

The transition t_0 is fired at $t_1=0$, reaching a new state S_1 , $c(t_0)=1$, $S_1=(P_1, T_1, X_1)$, and $P_1=(I_0(1))$, $I_1=(4,6,8)$, $X_1=(X_m)$;

If user press "modify speed" at $t_2=1$, accelerating the representation of the media(the parameter is 0.5), the new state is $S_2=(P_2, T_2, X_2)$, $P_2=(A_1(1))$, $I_2=(1,2,3)$, $X_2=(X_m)$;

Transition t_1 is fired at $t_3=2$, reaching the new state S_3 . $c(t_1)=1$, $S_3=(P_3, T_3, X_3)$, and $P_3=(A_1(1), V_1(1))$, $T_3=[10,15,20], [15,25,35]$, $X_3=0$. So the valid transitional period of time is $t_2=[10,15,20]$ and $t_3=[15,25,35]$. Because the type of t_3 is "Weak-And", t_3 fired after all input places (A_1, V_1) entering unlocked state. For some certain reasons, if A_1 or V_1 can't reach after t_1 fired, A_1 or V_1 is changed to unlocked state according to abnormal disposition after $\tau(t_2, A_1)$ or $\tau(t_2, V_1)$;

V_1) or $\tau(t_2, A_1)$;

If user press "skip" at $t_4=3$, firing of t_3 is skipped before the system fired it. $c(t_3)=1$, new state $S_4=(I_2(1), [12,22,32])$;

Transition t_4 is fired at $t_5=4$, reaching the new state S_5 . $S_5=(P_5, T_5, X_5)$, and $P_5=(V_2(1), I_1(1))$, $T_5=[15,25,35], [12,22,32]$, $X_5=0$. In S_5 , the type of transition t_5 is "Master". The interval of this type is determined by the valid period of time of "Master" input places. So t_5 can be fired in $[8,28]$, $c(t_5)=1$. Because the type of the synchronization is "Master", after fired there are 2 cases: the first, the returning road is from t_5 to t_1 ; the second, the returning road is from t_5 to t_1 . At the same time, because the "skip" contains a token in this place, the token of this place is deleted.

When V_2, I_1 are all unlocked, transition t_5 is fired. Reaching state $S_6=(P_6, T_6, X_6)$, and $P_6=(V_3(1))$, $T_6=[9,12,15]$,

$X_6=(X_p)$;

If the user press " pause " at $t_6=5$, it reaches state S_7 ,
 $P_7=(V_4(1))$, $T_7=[4,7,10]$, $X_7=(X_r)$;

If the user press " restart " , it reaches state S_8 ,
 $P_8=(V_3(1))$, $T_8=[4,7,20]$, $X_8=0$; t_6 fired after V_2 represented,
 $c(t_6)=1$.

Finally, the token reaches place of end, and the whole process finishes.

5. CONCLUSIONS

The Petri net is an important mathematic tool. It can describe and model the information processing system validly. To the concurrency, asynchronism and uncertainty of system, it also has dynamic analytic ability. In the process of research and applied development of Petri net, its applied area has exceeded the field of computer science. And it has been the better tool in researching the dynamic system of discrete events.

The distributed interactive multimedia synchronization model based on the temporal Petri net (ISPN) applies the type of dynamic synchronization, the temporal synchronization of inter-media and intra-media can be represented effectively and simply. On the controlling of the whole dynamic, the dynamic temporal behavior can be described integrally and accurately.

6. REFERENCES

- [1] Zhang Ming, Zhang Zhenglan, Research of Distributed Multimedia Synchronization Model based on Petri net, Mini-Micro System, Vol 22 No.6, June 2001.
- [2] Bai Yaobing, Study on Multimedia Synchronization Mechanism in Distributed Multimedia Systems, The Engineering and Application of Computer, July 2001.
- [3] Lu Xuanmin Wang Junxuan, Shi Haoshan, A Interactive Dynamic Synchronization Model of Multimedia based on Timed Petri net, The Electronic Technology of modern times, July 2002
- [4] Shan Zhiguang, Yang Yang, Extended Timed Stream Multimedia Synchronization Model based on Petr Nets, Computer Research and Development, 2000,37(2).
- [5] Sheng-Wei Guan, Sok-Seng Lim, Modeling with enhanced prioritized Petri nets: EP-nets, Computer Communications 25(2002)
- [6] Yahya Y.Al-Salqan, Carl K.Chang, Temporal Relations and Synchronization Agents, IEEE 1996
- [7] CovesC, CrestaniD, PrunetF, Design and analysis of workflow processes with Petri nets, In Proceedings of IEEE International Conference on Systems, 1998, 101~106
- [8] Donald A Adjero, Lee, Synchronization and User Interaction in the distributed Multimedia Presentation System, in Multimedia Database Systems, 1996:252~277



Lv Feng is a Full Professor and vice dean of Information Technology School, Wuhan University of Technology. He graduated from Wuhan University Technology in 1982; from Huazhong University of Science and Technology in 1989 with specialty of cybernation. He is a director of International Institute for General System Study, China sub-committees; director of Chinese Ceramic Society, Automatic sub-committees; standing director of Hubei Province Youth Science and Technology Society; director of Hubei Province electrician technology society; was a visiting scholar of Kanagawa University of Japan (1996). He has published two books, over 80 Journal papers. His research interests are in computer network communication, information system and information security technology, computer control and emulation; grey system theory and application.



Guo Yingli is a graduate student of Wuhan University of Technology. Her research interests are in computer network communication, information system and information security technology.

An Improved Genetic Algorithm for Solving QoS Distributed Routing Problem

Youwei Yuan¹, C.Cujaj²

¹Department of Computer Science and Technology,
Zhuzhou Institute of Technology, Hunan, China, 412008

E-mail:y.yw@163.com

²Department of Computer Sciences and TICAM
University of Texas, Austin, TX 78712

ABSTRACT

The distributed routing problem in computer networks is also known as the Steiner tree problem which has been shown to be NP-complete. In this paper, we propose a new QoS distributed routing algorithm based on Genetic Algorithms. We have incorporated the neural networks into our genetic algorithm (GANN) to dynamically control the rate of mating and mutation rate. Our algorithm considers multiple QoS metrics, such as bandwidth, delay, delay jitter, and packet loss rate, to find the multicast tree that minimizes the total cost. The analysis of the algorithm presented, backed up by simulation results, confirms its superiority over the other algorithms. This algorithm is simple, efficient, and scalable to a large network sizes.

Keywords: genetic algorithm, delay constrained, distributed routing, QoS.

1. INTRODUCTION

The QoS multicast routing (QMR) problem concerns the search of optimal routing trees in the distributed network, where messages or information are sent from the source node to all destination nodes, while meeting all QoS requirements. This problem is NP complete [1]. A number of efficient heuristic [4]-[6] or nature-based algorithms [7]-[13] have been proposed. In [13] and [16] a heuristic GA is used to solve the QMR problems. However, these approaches cannot be expanded. If one or more nodes are added into the network, the system needs to scan all nodes again to acquire the optimum solution.

In this paper, we present a novel heuristic distributed routing algorithm based on Genetic Algorithms. We have incorporated the neural networks into our genetic algorithm (GANN) to dynamically control the rate of mating and mutation rate. The Genetic Algorithms [10-12] provide robust and efficient search in complex spaces. Survival of the fittest “genes” and structured yet randomized genetic operations are the basic philosophies behind the algorithms. The main advantages of Genetic Algorithms include: (1) since solutions are coded as bit strings, referred to as chromosomes, large problems can be easily handled by using long strings; (2) genetic operations, such as crossover and mutation, are very easy to implement.

In this paper, we propose a method of getting near-optimal solution satisfying not only the QoS requirements but also as optimizing certain network resources such as bandwidth, end-to-end delay, in computationally feasible time, using the neural networks into our genetic algorithm to dynamically

control the rate of mating and the mutation rate. The article is structured as follows: Section 2, presents the problem description and problem formulation. Section 3 exposes the proposed multicast routing algorithm based on GA in combination neural network(GANN). Section 4 specifies the numerical results and conclusions and future work in section 5.

2. PROBLEM FORMULATION

Our goal: Using the GA to compute the candidate routes;

- (1) Constructing the neural network to dynamically control the rate of mating and mutation rate.
- (2) Optimize the candidate routes and get the optimal multicast tree.
- (3) Optimize the candidate routes and get the optimal multicast tree.

2.1 Genetic Algorithms

The genetic algorithms are typically implemented as follows:

Step 1: Initialize a population of chromosomes (solutions).

Step 2: Evaluate each chromosome in the population.

Step 3: Create new chromosomes by mating current chromosomes and apply mutation, and recombination when the parent chromosomes mate.

Step 4: Delete members of the population to make room for the new chromosomes.

Step 5: Evaluate the new chromosomes, and insert them into the population.

Step 6: If a stopping criterion is satisfied, then Stop and output the best chromosome (solution); otherwise, go to Step 3.

Crossover can produce offspring that are radically different from their parents. Suppose the crossover operation is performed on the two bit strings, “01110001” and “10011011”, and that they are split at the second bit; then, two new bit strings, “01011011” and “10110001” are generated (see Fig.1).

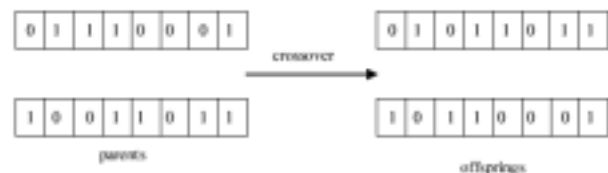


Fig.1. The Crossover Operation

Fitness function

The best solution for multicast M are selected in each iteration with the help of evaluation function F (also referred to as fitness function).

The fitness function is defined as follows[5]:

$$F = w_C F_C + w_D F_D + w_B F_B + w_L F_L + w_J F_J \quad (1)$$

Where F_c is the cost of the multicast route, F_D is for the delay requirements, F_B is for the bandwidth requirements, F_L is for the loss rate requirements, and F_J is for the jitter requirements. $w_i, i \in \{C, N, D, B, L, J\}$, are their respective weights. F_C is the primal fitness function as the main objective of multicast is to minimize cost. F_N is designed to punish a tree if it is illegal.

2.2 Network Model and Problem Definition

For a directed network graph, the problem of finding an optimal multicast tree with the least cost is formally defined as follows:

As far as multicast routing is concerned, a network is usually represented as a weighted diagraph $G=(V,E)$, where V denotes the nodes, and E , the set of arcs, corresponds to the set of the set of communication links connecting the nodes.

Consider a node u , with first deleted neighborhood $N'_1(u)$, degree $d(u) = |N'_1(u)|$, dominator $dom(u)$, and effective degree $d^*(u)$, where $d^*(u)$ is the number of its neighbors who have chosen u as their dominator. The core computation algorithm works as follows at node u .

1) Periodically, u broadcasts a beacon which contains the following information pertaining to the core computation:

$$(u, d^*(u), d(u), dom(u)).$$

2) If u does not have a dominator, then it sets $dom(u) \leftarrow v$, where v is the node in $N_1(u)$ with the largest value for $(d^*(v), d(v))$, in lexicographic order. Note that u may choose itself as the dominator.

3) u then sends v a unicast message including the following information:

$$(u, \forall_{\omega \in N'_1(u), (\omega, dom(\omega))}. (u, \{(\omega, dom(\omega)) | \forall_{\omega \in N'_1(u)}\})$$

v then increments $d^*(v)$.

4) If $d^*(u) > 0$, then u joins the core.

Definition 1: Delay-constrained least-cost multicast routing problem: Given a network $N(V,E)$, a source node $s \in V$, a destination node set $M \subseteq V - \{s\}$, the delay-constrained least-cost multicast routing problem is define as a multicast tree that satisfies:

$$\min \{cost(T(s, M)), T(s, M) \in T_f(s, M)\} \quad (2)$$

Where $T(s,M)$ is the set of all delay-constrained multicast trees constituted by s and M . It has been demonstrated that the delay-constrained least-cost multicast routing problem is NP-complete.

Definition 2: Given a network G , a source node s , destination node set R , a link delay function $D(.)$, a link cost function $C(.)$, and a delay bound Δ , the objective of the Delay-Constrained Steiner Tree(DCST) Problem is to construct a multicast tree $T(s,R)$ such that the delay bound is satisfied, i.e.,

$$\text{Delay}[ri] \leq \Delta \quad \forall_{ri \in R} \quad (3)$$

and that the tree cost $Cost(T)$ is minimized.

3. THE PROPOSED GANN-BASED ALGORITHM

3.1 The Proposed Multicast Routing Algorithm Based on GA in Combination With NN(GANN)

Fig.2 gives a clear view of flow of our designed algorithm.

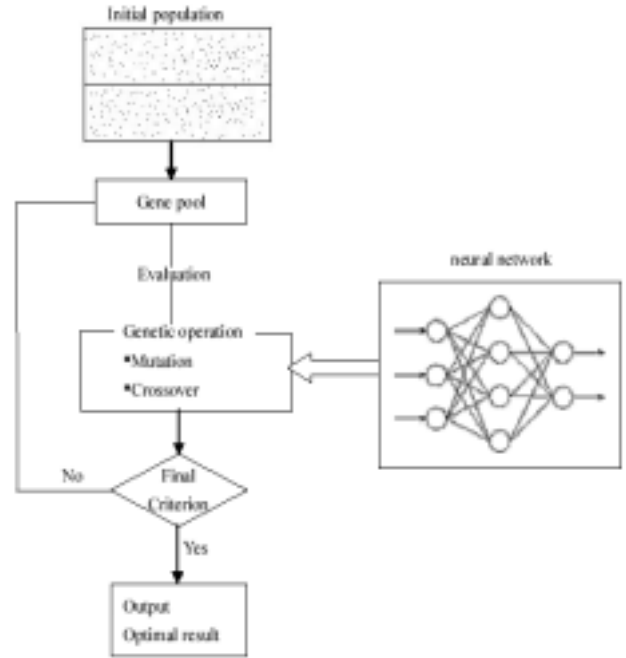


Fig.2 The Proposed Algorithm

3.2 Neural Networks For Controlling the Mating Rate and Mutation Rate.

It would be best if we just decrease the rate of mating, and increase the rate of mutation in an attempt to drive a process away from a local maxima. We use a feedforward neural network and train it using the error backpropagation algorithm. (As shown in fig.2.) At any state the input vector to the ANN, a_m , can be calculated using the function $CalculateInput()$. As shown in fig 3. The algorithm ensures that the value a_m correctly captures the variation in the fitness values for the previous ten iterations [15]. In our study we define the error function as the squared difference between the actual output obtained from the network and the desired output (desired output is the output given in the input-output pair). In case of mating rate, depending on the inputs we classify the desired output into four classes, namely excellent, good, average, and bad.

We apply the error function given to the desired output and the obtained output as follows:

$$E(m) = \frac{1}{2} \times \sum [b_k(m) - b'_k(m)]^2 \quad (4)$$

Where k ranges from one to the number of the output units, in our case five, and $b_i(m)$ denotes the i^{th} bit of the output vector of the ANN. Where $b'_i(m)$ denotes the i^{th} bit of the desired 5-bit output vector.

The ANN for calculating the mutation rate is same as we used to compute the mating rate.

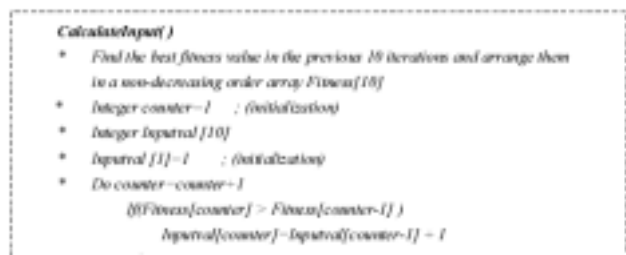


Fig.3. Calculate Input ()

4. PERFORMANCE ANALYSIS

4.1 Random Graph Generator

To investigate the performance of the proposed techniques we tested multicast configurations with as few as 20 nodes to as many as 200 nodes with varying levels of complexity. To guarantee fair results simulation results, we use the same graph generator (Waxman 1988)[9] that is used in all problems related to multicasting. GANN has been implemented in C++. The experiments were carried out using the Arpanet topology and sparsely connected random graphs. The experiment was run repeatedly until confidence intervals of less than 5%, using 95% confidence level, were achieved for all measured quantities. On the average, 320 different networks were simulated in each experiment in order to reach such confidence levels. At least 260 networks were simulated in each case.

4.2 Compare the Performance of the GANN with the Other Heuristics

For the purpose of evaluating the efficiency of the proposed routing method. In addition to our protocol GANN, three other QoS multicast routing protocols (QoSMIC[3], Sun-GA[13] and QMPR[4]) are also simulated under the same network configuration for performance comparisons. For QoSMIC, local search and tree search are implemented as parallel procedures; directivity, local minima, and fractional choice [4] are also implemented. Sun-GA has better performance in terms of number of messages, time and success rate, while produces equal quality of multicast trees in terms of the tree cost. QMRP is extended by us in the simulation to support additive metrics, such as delay bound. In QMRP, the maximum branching degree (MBD) is set to five, and the maximum branching level (MBL) is unused. For the above three protocols, a unicast routing protocol providing the shortest path in terms of hops is assumed to exit. Since the unicast protocol constitutes part of the QoS-based routing, it consumes network resources and contributes to routing overhead as well.

Three performances metrics, success ratio, average message overhead and average connection time are used in the examination.

4.2.1 The Routing Request Success Ratio

Fig.4. compares the failure rate for different protocols, and our protocol always has the lowest failure rate possibility. This can

be reasoned as follows. Our protocol, on the other hand, takes a simpler but more effective approach. It aggressively searches all the available feasible path. As a result, it has the highest routing success ratio.

4.3.2 Tree Overheads and Execution Time Versus Network Size

Fig. 5. compares the routing overhead for different protocols. As expected, it makes sense to be more aggressive in routing (hence more overhead) to ensure a better chance of successful routing.

Fig.6. shows: with the adding of the number of nodes, the operation times of GANN is not much added. GANN is the fastest than the other three optimal algorithms.

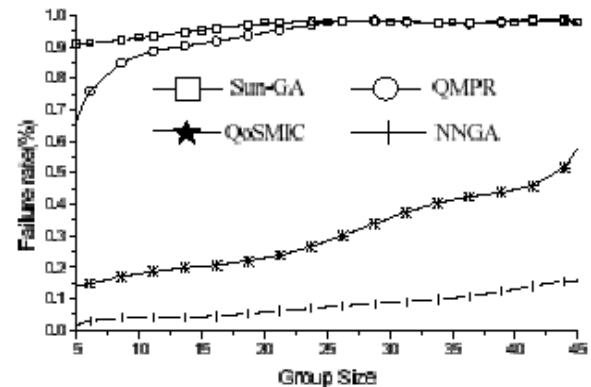


Fig.4.(a) Failure Rate Versus Delay when Group Size=45

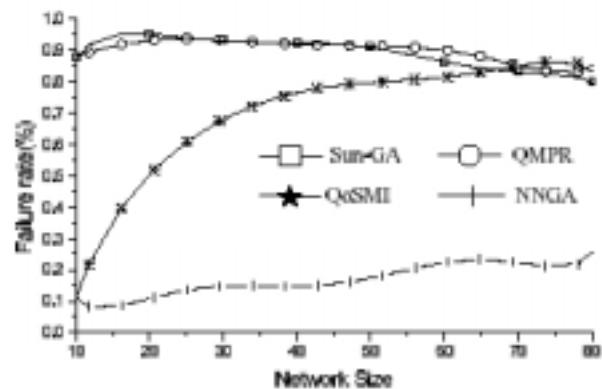
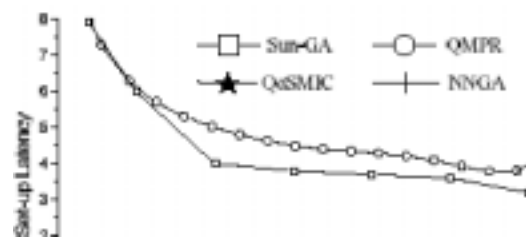
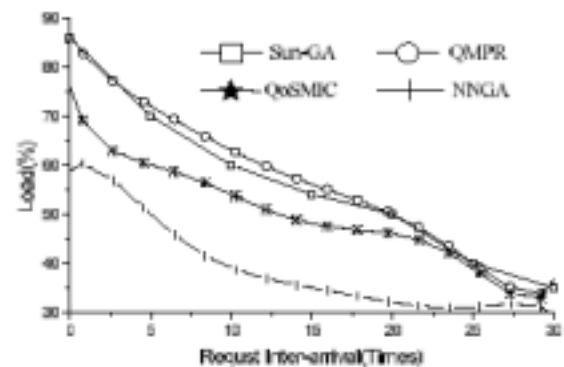


Fig.4.(b) Failure Rate Versus Delay when Group Size=80



There are of course other methods such as fuzzy-control and genetic algorithms which we would like to investigate in future work.

6. ACKNOWLEDGEMENTS

The work was supported by nature science foundation of China (No.50274080) And also supported by the academic science foundation of Hunan province (No.02C643)

7. REFERENCES

- [1] N. Banerjee, S.K.Das. "Fast determination of QoS-based multicast routes in wireless networks using genetic algorithm".. IEEE international conference on communications,. Volume(8),pp:2588-2592. 2001.
- [2] K. Carlberg, J. Crowcroft, "Building shared trees using a one-to-many joining mechanism," ACM SIGCOMM Computer Communication Review, vol. 27, no.1, pp. 5-11. Jan.1997.
- [3] M. Faloutsos, A. Bamerjea, and R. Pankaj, "QoS MIC: Quality of service sensitive multicast internet protocol," in Proc of ACM SIGCOMM'98, pp.144-153. Sep.1998.
- [4] S. Chen, K. Nahrstedt, and Y. Shavitt, "A QoS-aware multicast routing protocol." IEEE Journal on Selected Areas in Communications, vol.18, no.12. December 2000, pp.2580-2592.
- [5] Q. Gu, C. Chu. "Solving the QoS multicast routing problems using genetic algorithms." "Soft computing and telecommunications." Springer-Verlag, 2003.
- [6] Y. Yuan, L. Yan. "A heuristics genetic algorithm for distributed multicast routing". Malaysian journal of computer science. Vol(15), no.2 pp:70-77, 2002.
- [7] Z. Wang, J. Crowcroft, "Quality of service for supporting multimedia applications", IEEE Journal on Selected Areas in Communications, Vol(14), no(7), pp.1228-1234. 1996.
- [8] X. Yao, "Evolutionary artificial neural networks." International Journal of Neural Systems, Vol(15), no(4), pp:203-222. 1993.
- [9] Waxman B.M., "Routing of multipoint connections", IEEE TSAC, Vol(9), no(6), pp: 1617-1622. 1998.
- [10] T.T. Chow, Z. Lin and C.L. Song. "Applying neural network and GA in chiller system optimization" In Proc of 7th International IBPSA conference" pp: 1059-1066. Brazil, 2001.
- [11] D. Schaffer, D. Whitley, D. and L. Eshelman, (1992) "Combinations of Genetic Algorithms and Neural Networks: A survey of the state of the art". In Proc of the International Workshop on Combinations of Genetic Algorithms and Neural Networks. pp: 1-37. 1992.
- [12] B. Julstrom. "The code: a better string coding of spanning trees for evolutionary search." In Proc of 2001 Genetic and Evolutionary Computation Conference Workshop Program". pp:158-162. 2001.
- [13] Q. Sun, "A genetic algorithm for delay-constrained minimum-cost multicasting. Technical Report, IBR, TU Braunschweig," Butenweg, 74/75, 38106, Braunschweig, Germany, 1999.
- [14] A. Roch, A. Orda. "QoS routing in networks with inaccurate information: Theory and Algorithms" IEEE/ACM Transactions on neural networking. Vol(7), no.3, pp:350-364. June 1999.
- [15] A. Muthukaruppan, S. Suresh. "An artificial neural guided parallel genetic approach to the routing problem for field programmable gate arrays." www.isrl.uic.edu/~amag/langev/paper
- [16] D. Goldberg. "Genetic algorithms in search, optimization & machine learning" Addison-Wesley, New York. 1989.

Fig.5 Comparison of Routing Overhead

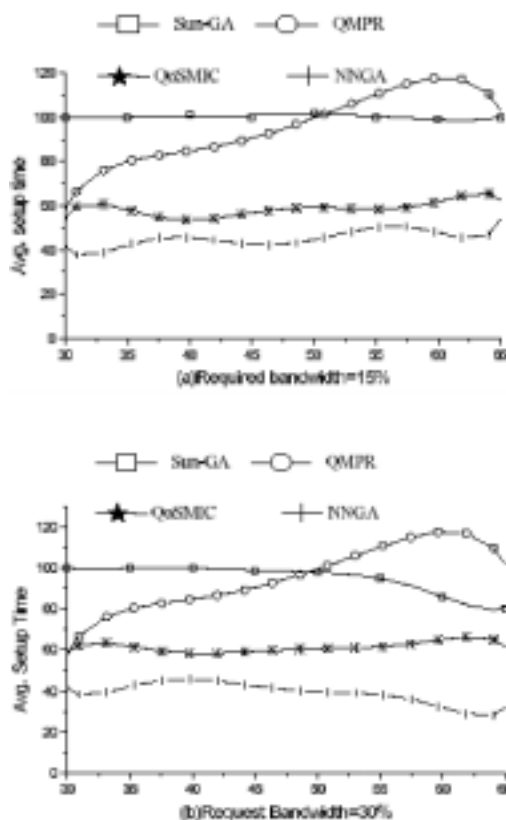


Fig.6 Average Setup Time vs. Network Load (delay bound=120)

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a new multicast tree selection algorithm to simultaneously optimise multiple QoS parameters which are based on a Genetic Algorithm in combination with a neural network(GANN).

Compared with the other optimal algorithm, the proposed algorithm gives better performance in terms of the success rate, the tree cost, the number of exchanged messages and the convergence time..

In conclusion, GANN is well suited for QoS-based multicast routing. With this algorithm, a dynamic constructed multicast routing tree which has a near optimal network cost under the delay bound constraint can be constructed in a real time manner.

Design and Implementation of an Embedded VPN Gateway Based on IPSec

Zheng Yuanjiu, Liu Quan, Li Fangmin

School of Information Engineering, Wuhan University of Technology
Wuhan Hubei, China 430070

E-mail: zyj@mail.whut.edu.cn Tel: +86 (0)27 87299825

ABSTRACT

In this paper, a design solution of embedded VPN gateway based on IPSec is proposed, which works on the cooperation of double CPU processors. The MPC8250 chip of Motorola Inc. is selected as the main network process CPU. The TMS320C6202 DSP of Texas Instruments is selected as the processor to do encryption and decryption works. This solution resolves the transmission speed bottleneck of Wide Band bases on the co-operation of double CPU processors. It ensures the safety of data transmission by hardware encryption. This paper puts emphases on the introduce of the hardware architecture of the VPN gateway, and then it analysis the function of each module.

Key words: VPN embedded system IPSec FPGA

1. INTRODUCTION

VPN^[1] (Virtual Private Network) is often defined to build up a temporary safety connection on the public network. It is a safety and stable tunnel through the disordered public network. The VPN builds up a secret and private network tunnel through the public network, protects this network not to be limited by the distance. Simultaneously in the VPN the visit between each point should be the same with traditional network, it helps the long-distance user, the company branch office, the commercial partner and the supplier with company's internal net to establish a credible safe connection, and guarantee data transmission safely. Because of the low cost of the Internet public facility, people who once used expensive special telephone line hope using the this more inexpensive method.

In building up such a VPN tunnel, the key factor is to design a gateway, which can realize the function of VPN, and that, the gateway itself is safe. With the progress of network equipments, at present the bottleneck on the research of VPN gateway is how to satisfy simultaneously the safety of VPN and the high speed of data transmission. So we design one kind of double CPU gateway based on embedded and VPN IPSec. It improves the transmission speed by the cooperation of double CPU and strengthens the data safety by hardware encryption system. This kind of gateway has wide application foreground in the present high-speed network.

2. IPSEDC PROTOCOL AND EMBEDDED SYSTEM

2.1 IPSec Protocol

At present the main work of VPN is data packing on the network layer, which is packet data according to the third tunnel protocol. The main protocol in the third layer is the

IPSec^[2] protocol. The IPSec protocol is IP security standard published by TETF, it integrates several kind of security technologies to form a more complete system, and it is a standard protocol used in authentication, privation protection, and integrity. The IPSec protocols use the password technology to ensure the data security from three aspects, including the user authentication, the integrity check and data encryption. The authentication is used to authenticate the identity of the host and other users. The integrity check is used to ensure that the data has not been changed when it is transmitted on network. The encryption is used to encrypt the IP address and the data to guarantee the privation.

Ever since IPSec protocol standard has been published, it has received much attention and support of the most manufacturers. By now most of the commercial products about VPN appeared are network security products based on the IPSec tunnel protocol, including software and hardware products. The CISCO Inc. and Intel Corporation have different series of the VPN solution products. Here our solution of the gateway also is based on the IPSec tunnel protocol.

2.2 EMBEDDED VPN GATEWAY

Embedded VPN gateway means that the VPN gateway is designed based on the hardware platform of embedded operation system. Considering the following factors, we choose embedded system platform to realize the VPN gateway but not based on common platform.

Security

The VPN gateway is a device to protect data security. To realize this, so the gateway itself must be safety. Otherwise it is of no use regardless how strong the data passed through this device is encrypted and authenticated. Different from common computer platforms, embedded system is designed to service the applications. Both of its hardware and software are designed according to the application characters, so the software and hardware system are predigested in the greatest degree. On other sides, embedded system is different from single user or multi-user system. It usually executes a series of fixed programs and seldom imports new programs, so there are no spite user problems.^[3] Therefore the system platform is absolutely safety when designing the gateway based on embedded system platform.

Reliability

The reliability is the running stability of the system itself. Most device factors, such as the running environments, loads etc., are considered synthetically in embedded system designing. So some measures, such as devices preventing interferers and hardware watchdog, are added. What's more, the comparative simple of the system platform also reduce the number of device equipments. All of these lower the probability of failure emergence.

Ratio of P/C

The comparative function fixity of the VPN gateway determines the fixity of its support platform. When designing the gateway based on embedded platform only the necessary function parts are kept down, so the cost is the lowest. And that the function of VPN is lossless in minimized designing.^[4] The design of embedded system conforms to the principle of that the function requirements are priority. The special requirements have been considered when design embedded system platform. So this kind of design need no extend interfaces, different from common hardware platform, it has no lose to the performance.^[5] Therefore the design based on embedded system platform has higher P/C ratio than that based on common platform.

3. HARDWARE DESIGN OF EMBEDDED VPN GATEWAY

As shown in Figure 1, the whole hardware system is consist of the main CPU network process module, the DSP cooperated process module, the encryption process module and other necessary interfaces and devices, such as RAM, ROM, nvRAM, Ethernet interface and serial interfaces etc..

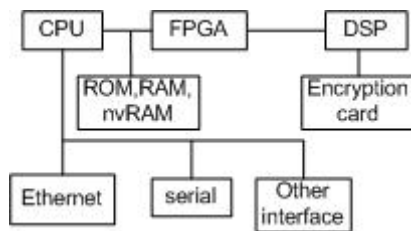


Figure 1 Consists of system

The speed of data transmission in a perfect gateway should be linear. But in the VPN gateway, the process of data packets not only includes the policy matching and auditing, once the VPN policy is needed, it also includes the data parity checkout, encryption process and decryption process. The work of computing is so huge that it is the key factor which affects performance of the VPN gateway. So in this paper, we use double CPU processors to realize the hardware computing platform. The main CPU processes common tasks, and the cooperation DSP processes some special work which cost much computing resource, such as encryption and decryption.

There are two CPU processors on the same embedded system platform, so the following describes how they work together.

The main CPU sends the data which needs encrypted to the SDRAM, and notices the FPGA device that the data has been sent by the control bus. The FPGA gets the control signal and reads the data in SDRAM, then it finishes the logic transform task and sends the data to DSP or encryption card. After the DSP or encryption card finished their encryption or decryption work, they send the encryption data back to FPGA. Also after the FPGA finishing its transform work, it sends the encryption data back to SDRAM, and sends the control signal to the main CPU. Then the main CPU read the data processed by DSP from the SDRAM, and finishes the relative protocol processing. If there is extend special encryption sub-card, then the process of DSP is bypassed, and encryption work is done by the sub-card

directly.

3.1 The main CPU network process module

This module is the kernel module in the VPN gateway. It realizes the route function of the gateway and the IPSec protocol. The type of the main processor is MPC8250, it is a special communication processor made in Motorola Inc. This module has three 100M Ethernet interfaces for connection use, two serial interfaces for downloading configuration and on JTAG interface for debugging. It is responsible for the most function of a gateway, including reading data from the physical layer, processing data according the IPSec protocol standard, and send the data needs to be encrypted to other modules through bus.

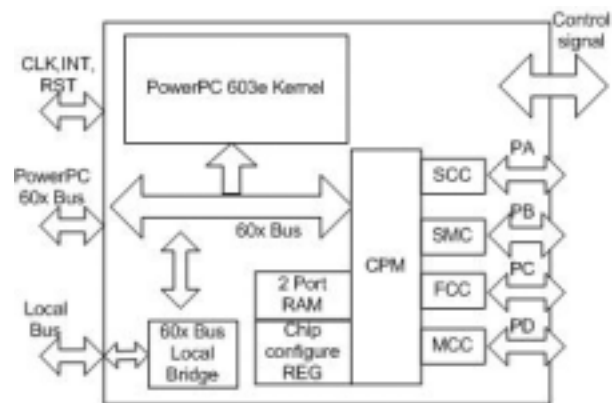


Figure 2 Structure of MPC8250

As shown in Figure 2, the chip of MPC8250 is the most advanced communication micro-processor designed for telecom and network markets. It has high performance 603 kernel of embedded PowerPC, works at the frequency from 100 to 266MHz. This type of processor has several kinds of bus, including local bus and 60x-bus. We use 60x-bus as the work bus and the bus frequency is 66MHz.

3.2 DSP co-processor module

The main work of DSP co-processor module is on the asymmetry encryption algorithms. There is no special module with asymmetry encryption algorithms, so we use DSP processor to finish these algorithms such as RSA^[6] etc. The encryption algorithms, like modular arithmetic, cost much CPU resource, so it is very necessary to use DSP as the encryption co-processor. We use the chip of TMS320C6202 DSP made in Texas Instruments as the co-processor.

3.3 Encryption and decryption module

Encryption and decryption module is a separate sub-card system. The main function of this module is to realize the hardened symmetry encryption algorithms of 128 bits such as DES, triple DES etc. The sub-card module is divided in two function units. One unit is the logic interface of PCI bus based on FPGA, the other one is the encryption algorithms based on FPGA. The performance of PCI interface ensures that it doesn't affect the high speed of the whole system, and the plug and play character improves the expansibility of the whole system. The hardware realization of encryption based on FPGA could ensure the high encryption speed and enough strong intension.

4. ARCHITECTURE OF SOFTWARE SYSTEM

Both Linux and VxWorks are good embedded system platform, but VxWorks system is based on micro kernel structure, all of its work is in the way of tasks except the task schedule, and it can realize IPSec protocol by change the older protocol stack. So considering the efficiency and real-time character, VxWorks system is better than Linux.

To realize the function of VPN, the protocol stack with IPSec function should be load to the VxWorks system platform. So the important work is how to realize IPSec.

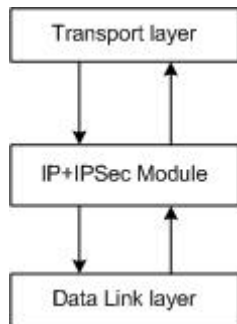


Figure 3 Implementation of IPSec

As shown in Figure 3, a way of BUMP-IN-STACK is used to combine the IPSec with IP protocol. The IP data packet will be checked with IPSec policy firstly, then according the rules it will be discarded, bypassed or added policies. If it needs to be added policies, then relative process will be done to it. Otherwise, it will be discarded or processed as common IP packets. And at last it will be routed.

The importance of IPSec function can be realized by change the TCP/IP protocol stack on VxWorks system source code. According to the route table to choose the transmit interface of the data packet received from the Ethernet card. If the packet should be sent to local host, then it will be sent by a virtual interface. Before sending the packet, the packed should be checked if the packer is to input or output according the aim network address, and then process it relatively. When the data packet is sent to the VPN gateway based on IPSec, the network layer will judge which Ethernet card it should be sent from the route information. If it is sent to the local virtual Ethernet interface, it means the packet is an input packet. Or if it is sent to other physical interfaces, and that means it is an output packet. Then process according relative protocol will be done. And the packet which has been processed will be sent back again to the route process module.

5. TEST RESULT AND ANALYSIS

As shown in Figure 4, it is our test environment. By all kinds of tests, we can get the results as shown in table-1.

As shown in table-1, the encryption speed of hardware is about 5 to 6 times faster than that of software. In fact the speed of separate hardware encryption module can reach line speed processing ability. Thus the encryption speed is not the bottleneck of the VPN gateway anymore. With more high speed of CPU, the G-bits speed of gateway is reliable.

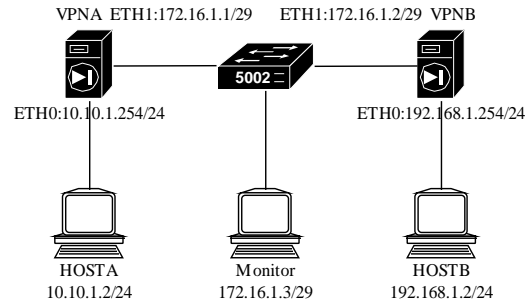


Figure 4 Test environment

Table -1 Test Result

Engine Algorithm	Software Crypto (KB/s)	Hardware Crypto (KB/s)
DES, MD5	22~22.3	94.5
DES, SHA1	19~19.5	101.7
3DES, MD5	9.4~9.5	47.25
3DES, SHA1	8.8~8.9	49

6. CONCLUSIONS

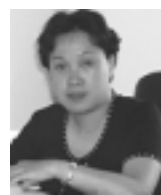
Based on the analysis of speed and security problems in data transmission over wide band network, this paper gives out a solution of embedded VPN gateway based on the co-processing of double processors. The co-processing of double CPUs improves the speed greatly and the hardened encryption based on FPGA provides very high encryption speed and enough strong intensity.

7. REFERENCES

- [1]. B.Fox, B.Gleeson. Virtual Private Networks Identifier.RFC-2685 Sep 1999.
- [2]. Kent S, Atkinson R. Security Architecture for the Internet Protocol. Request for Comments. RFC 2401, IETF, Nov 1998
- [3]. Steve Furber., Tian Ze translate, Architecture of ARM SoC, Beijing: Beihang Publication Inc., Sep 2002. (in Chinese)
- [4]. Coste P., Hessel F., Jerraya A. A Multilanguage co-design using SDL and mat-lab in SASIMI 2000, Kyoto, Japan, 2000 April.
- [5]. Patterson David A, Hennessy Jhon L. Computer Architecture A Quantitative Approach. Second Edition, Morgan Kaufmann.1996
- [6]. Bruse Schneier, Wu Shizhong translate, Applied Cryptography: Protocols, Algorithms, and Source Code in, Industry of machine public Inc., Jan.2000.(in Chinese)



Zheng Yuanjiu, male, born in 1980.1, is the master of communication and information system, Wuhan university of technology. The study field is on embedded system and network security.



Liu Quan, female, full professor, tutor of doctor, is the dean of the school of information engineering, Wuhan university of technology, her research interests are information security, signal processing, communication technology, grid computing and network security.

Shortest-Path Routing Based on Ant-Algorithm

Min LianYing, Yang JinYong

School of Computer Science and Technology, Wuhan University of Technology
Wuhan, Hubei, 430063, China

Email: min_ly@mail.whut.edu.cn, Yang_jin_yong@sohu.com

Tel: +86 (0)27 13871140872, +86 (0)27 13554071386

ABSTRACT

In this paper we first analyze the theory of ant algorithm and its math model, then we put forward a novel approach to solve the shortest-path-routing problem that uses the ant algorithm. Finally we set up an experiment to testify the validity and efficiency of our approach.

Keywords: Ant algorithm; The Shortest-Path; Routing

1. INTRODUCTION

Routing is one of the important problem in the field of the packet-switched computer network.. A classical problem about routing is to figure out the shortest path between two switch-nodes, the measure to weight the shortest path can be such as the delay, the number of switch-node, the function value of multiple link parameters and so on, therefore the shortest path algorithm has a comprehensive significance. There are two famous algorithms to calculate the shortest path, i.e., the Dijkstra algorithm and Bellman-Ford algorithm, both of which have a time complex of polynomial. With the demanding of ever-shorter computing time, these years some researchers have developed some method based on such algorithm as the Neural Network algorithm [5], the Genetic algorithm [6], although these algorithms have fast stochastic global searching capabilities, they could not use the feedback information of the computing procedure, they tend to do a lot of iterations, which results in the low efficiency and precision. The ant algorithm, put forward by Italy scholar Dorigo, make full use of the feedback information by a material-pheromone excreted by the ants, to achieve the goal of global optimization. The ant algorithm are featured as follows:

- I. The algorithm is based on a positive feedback mechanism; it converges the ultimate optimized results through the accumulation of pheromone.
- II. The algorithm is a global optimization method that can solve not only the onefold optimization problem, but also the problem of multiple constraints.
- III. The algorithm is a distributed optimization method, so it can be implemented by parallel computing [3].
- IV. The algorithm can search the new path adaptively if the parameters of the system are changed, so it can achieve the goal of finding the optimized result dynamically.

By exploiting the above merits of ant algorithm, researchers have been able to design a number of successful algorithms in such diverse field as combinatorial optimization, multi-robot system, graph drawing and partitioning, and so on. [1].

2. THE THEORY AND MATH MODEL OF ANT ALGORITHM

2.1. The Theory of Ant Algorithm

Studies show that the ants can find the shortest path between their nests and a food source. This capability is attributed to a volatile chemical material-pheromone. Individual ants deposit pheromone on the trail while walking, and the other ants follow the pheromone trails with some probabilities which are proportioned to the density of the pheromone. The more ants walk on a trail, the more pheromone deposited on it, and more and more ants follow the trail, thus a positive feedback mechanism comes into being. Through this mechanism, ants will eventually find the shortest path. Fig. 1 shows exactly how the ants find the shortest path.

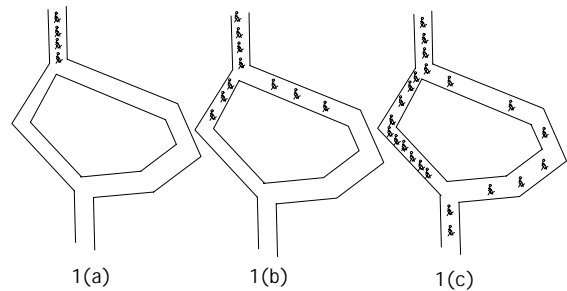


Fig.1 The sketch map of the ant theory

At first (Fig. 1(a)), the ants select randomly one of the branches with equal probability for there is not any heuristic information. So the result is that half of the ants choose the left branch, half of the ants choose the right branch, and each individual deposits pheromone respectively on the trail it passed (Fig. 1(b)). Suppose that all the ants have the same speed, we can figure out that intensity of the pheromone of the short branch is higher than that of the long branch, because of the volatility of the pheromone. Then the shorter branch attracts more ants to choose it, which in turn continue to enhance the intensity of the pheromone. Through this positive feedback mechanism, after a while, the density of the pheromone of the shorter branch will highly exceed that of the long branch, and the newly coming ants will choose the shorter branch with a greater probability. In the end, most of the ants will follow the shorter branch (Fig. 1(c)).

2.2. The Math Model of the Ant Algorithm

Let's suppose that all the information about the trail is stored in a graph G .

2.2.1. The modification of the pheromone

Suppose at iteration t , the current pheromone intensity value of edge (i, j) is $\tau_{ij}(t)$. After completing its tour $T_k(t)$, the k th ant lays a quantity of pheromone $\Delta\tau_{ij}^k(t)$ on edge (i, j)

belong to $T_k(t)$, $\Delta\tau_{ij}^k(t)$ is a function of the length L_k of tour $T_k(t)$ [1]:

$$\Delta\tau_{ij}^k(t) = \begin{cases} Q/L_k & \text{if } (i, j) \in T_k(t) \\ 0 & \text{if } (i, j) \notin T_k(t) \end{cases} \quad (1)$$

where $T_k(t)$ is a set of each edge (i, j) , L_k is the total sum of all the length of edge (i, j) .

Q is an adjustable parameter.

After each ant completes its tour between the source vertex and the destination vertex, at the next iteration $t+1$, the pheromone of each edge (i, j) is given by [1]:

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (2)$$

where ρ is a coefficient of evaporation of pheromone $0 < \rho \leq 1$.

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t), m \text{ is the number of ants.}$$

2.2.2. The probability of choosing an edge

The probability $P_{ij}^k(t)$ with which an ant chooses a edge (i, j) to move is determined by: [1]

$$P_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in J_k(i)} [\tau_{il}(t)]^\alpha [\eta_{il}]^\beta} & \text{if } j \in J_k(i) \\ 0 & \text{if } j \notin J_k(i) \end{cases} \quad (3)$$

where η_{ij} is the heuristic visibility of edge (i, j) , generally it is given a value of $1/d_{ij}$, d_{ij} is the length of edge (i, j) . $J_k(i)$ is a set of vertex which remain to be visited when ant is at vertex i . α and β are two adjustable parameters that control the relative influences of pheromone $\tau_{ij}(t)$ and heuristic visibility η_{ij} . If $\alpha=0$, the closed vertex are more likely to be selected; this respond to a classical stochastic greedy algorithm. If on the contrary $\beta=0$, only pheromone amplification is at work: This method will lead the system to a stagnation situation, i.e., to a situation in which all the ants generate a sub-optimal tour. So the trade-off between edge length and pheromone intensity appears to be necessary [1].

3. FIND THE SHORTEST PATH ROUTING USING ANT ALGORITHM

3.1. The Network Model of Shortest Path Routing

The packet-switched network can be defined as $G=(V, L, W)$, as showed in Fig.2, a node represents a switch node, V is a set of switch node, the edge connects two nodes represents a data link, L is a set of data-link, W is the set of the cost of data link, Let T be the path from the

source node S to the destination node D , $f(T)$ is the cost expensed on T , then our goal is to find a path which can make $f(T)$ the minimum.

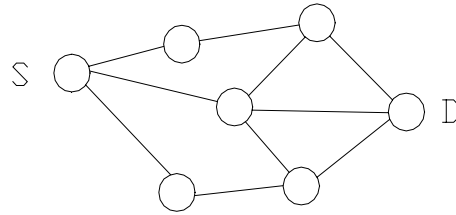


Fig. 2 The network model of shortest path routing

3.2. Find the Shortest Path Routing in Graph G Using Ant Algorithm

In order to have a good convergence, we give the following constraints:

- I. The value of pheromone on each edge in graph G is initialized with a maximum value τ_{\max} .
- II. To avoid converging at a non-optimal solution, the pheromone of each edge (i, j) is confined within $[\tau_{\min}, \tau_{\max}]$, the pheromone will be set to τ_{\min} or τ_{\max} constrainedly if it is beyond the boundary [4].
- III. Each node will be visited once and only once.
- IV. At each iteration, only those ants who arrive at the destination node successfully deposit pheromone on each edge.

Based on the above constraints, the algorithm to find the shortest-path routing is the following: (Fig. 3)

```

set  $L_0$ , the initial value of the shortest-path length,
equal to a maximum value( e.g., 99999);
set  $u_0$ , the node set of the shortest-path, equal to an
empty set;
initialize the pheromone of each edge  $(i, j)$ ;
for( $t=1$ ;  $t < \text{iterationCount}$ ;  $t++$ )
{
    for( $k=1$ ;  $k < \text{antNumber}$ ;  $k++$ )
    {
        set  $i$ , the current position of ant  $k$ , equal to the
        start node of  $S$ ;
        set  $u$ , the node set of current path, to have
        only one element of  $S$ ;
        while (ant  $k$  doesn't arrive at destination node
            D and a feasible successor node
            exists)
        {
            select successor node  $j$  with
            probability  $P_{ij}^k(t)$ ; //Eq(3)
            add node  $j$  to the node set  $u$ ;
            set  $i$ , the current position of ant  $k$ , equal to
             $j$ ;
        }
        if ( ant  $k$  arrives at destination node D

```

```

        successfully)
    {
        compute  $L$ , the length of the path ant  $k$ 
            walked at iteration  $t$ ;
        compute  $\Delta\tau_{ij}^k(t)$ , the pheromone
            deposited on each edge by ant
             $k$ ; // Eq(1)

        if ( $L < L_0$ )
        {
            set  $u_0$  equal to  $u$ ;
            set  $L_0$  equal to  $L$ ;
        }
    }
    do pheromone update; // Eq(2)
}

```

Fig. 3 The solution to find the shortest-path routing based on ant algorithm

When the algorithm is completed, u_0 is the path that we want, and the length of the path is L_0 .

4. SIMULATION TEST

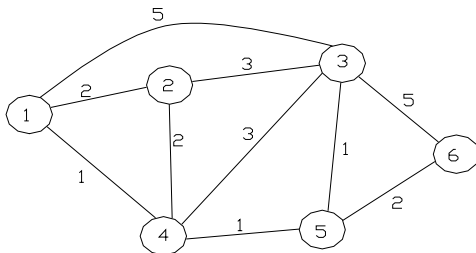


Fig. 4 The network on which we took a simulation test

In order to testify the validity and efficiency of our algorithm, we set up a experiment. the network is shown in Fig. 4, the number of each edge represent the cost. The parameters in the simulation test are: $\alpha = 1$, $\beta = 5$, $Q=100$, $\rho=0.5$, the initial value of pheromone on each edge is set to $\tau_{\max} = 1000$, and $\tau_{\min} = 100$. the result is as following:

Table 1 result of the test

Routing request (S, D)	Routing	Cost
(1, 6)	1 4 5 6	4
(2, 6)	2 4 5 6	5
(3, 6)	3 5 6	3

5. CONCLUSION

In this paper we designed a solution to solve the shortest-path routing problem. Still there are some question needed to be addressed and solved, for example, how do we set the proper value of the parameters? How can we accelerate the convergence of the algorithm? We are glad to see that some research have been done concerned with these issues [2,3], we think future research work should continue to deal with them. What we want to point out is that, done to the flexibility of ant algorithm, it has shown remarkable vitality in network routing, e.g., researchers have used ant algorithm in the fields of QoS routing and multicast routing[7,8]. We are convinced that ant algorithm will bring more fruitful research result in network routing with more and more people are interested in this field.

6. REFERENCES

- [1] Marco Dorigo, Eric Bonabeau, Guy Theraulaz. Ant algorithm and stigmergy. Future Generation Computer Systems 16(2000) 851-871
- [2] Walter J. Gutjahr. ACO algorithms with guaranteed convergence to the optimal solution. Information Processing Letters 82(2002) 145-153
- [3] Marcus Randall, Andrew Lewis. A Parallel implementation of Ant Colony Optimization. Journal of Parallel and Distributed Computing 62(2002) 1421-1432
- [4] Thomas Stützle, Holger H. Hoos. MAX-MIN Ant System. Future Generation Computer Systems 16(2000) 889-914
- [5] Hu Shiyu, Xie Jianying. Shortest-Path Routing Using Neural Network. Communication Technology 2003, 140(8), 45-47
- [6] Sun Q. A. Genetic algorithm for delay-constrained minimum-cost multicasting. Technical Report, IBR, TU Braunschweig, Germany, 1999
- [7] Zhang su-bing, Lu Guo-ying, Liu ze-min, Zhou zheng. QoS Routing Based on Ant-algorithm. Journal of circuits And Systems, 2000, 5(1)
- [8] Li Shenghang, Pan Li, Zhu Hongwen, Liu Zemin. Ant-algorithm Based Multicast Routing. Computer Engineering. 2001, 27(4), 63-65



Min LianYing is an associate professor in School of Computer Science and Technology, Wuhan University of Technology. His research interests are in software engineering and computer network technology, now he is devoted to embedded system.



Yang JinYong is a master graduate in School of Computer Science and Technology, Wuhan University of Technology. He is interested in AI and computer network technology.

Data Communication between Monitor computer and PLC Based on the Profibus

Liu Qing, Guo Jianming, Wang Yanwen
School of Automation Wuhan University of Technology
Wuhan, Hubei, P. R. China
Email: qliu2000@mail.whut.edu.cn Tel.: 027-62038303

ABSTRACT

In networked control system (for short, it's called NCS) the detectors, controllers, monitor computers, servers, human-machine interfaces (for short, it's called HMI) and etc. are all usually distributed, the communications in them are needed in some way. The data communication in real-time and reliability is most important technology in NCS. Geotextiles-Laying Vesse NCS introduced in the paper consists of Siemens s7-300 series Programming Logic Control (for short, it's called PLC) and industrial computer mainly. The hardware configuration and Profibus-DP communication technology are discussed. The DLL files provided by Siemens PRODAVE software are analyzed and used to realize communication between PLC and monitoring computer. The communication program in Visual Basic and Visual C programming language is presented. The communication way is successful used in Geotextiles-Laying Vesse. The result shows that data communication's error code rate is lower and communication rate is higher.

Keywords: Network; Fieldbus; Communication; Programming Logical Controller.

1. INTRODUCTION

Networked Control System, that is control network, results in the development and amalgamation of computer, communication and control technology. It incarnates the trend that control system will develop towards network, integration, distribution and node intelligentized. Distribute Control System (in short, it's called DCS), Fieldbus Control System (in short it's called FCS) and industrial Ethernet are all belong to NCS. In conventional control system information is detected by center control unit and control commands are also send out by center control unit. So the system's topology is simple. In NCS detectors, controllers, monitor computers or human-machine interfaces, servers, and etc. are all usually distributed. The communication in them in some way is needed and the communication in real time and reliability among them is the precondition that the NCS can work reliability. In the development of a real project of Geotextiles-Laying Vesse NCS, by using Visual Basic and Visual C programming language the program design ways of communication between monitor computer and SIEMENS s7-300 PLC are studied especially via PROFIBUS based on the PRODAVE S7 software, which is offered by SIEMENS Corp.

2. THE GEOTEXTILES-LAYING VESSE NCS STRUCTURE

The Geotextiles-Laying Vesse NCS consists of four suits of Siemens s7-300 PLCs, an IPC, a HMI and some sensors. The process signals are detected and control commands are sent by PLCs. IPC is used as host monitoring computer and to get GPS and sounder signals. The network structure is based on Profibus-DP. The system's structure and hardware configuration are showed in Fig1.

In Fig1. a suit of s7-300, which model is CPU315-2DP and has two Profibus-DP ports and a MPI port, is used as DP network master. The other three suits of s7-300 PLCs, which model is CPU314, are used as slaves in control network.. It is the single-master and multi-slave structure. There is a CP342-5 Profibus-DP block in every slave. There is a CP5611 block in IPC to support MPI and Profibus communication. TP27 in Fig.1 is one of the Human-Machine Interface (for short, it's called HMI). The system's topology is bus. In this system the communication among them is the key to insure system's work reliable.

Profibus-DP, which is used in the control network, is a kind of international standard bus. The most transmission rate is 12 Mb/s. When the transmission rate is set at 9.6Kb/s, the most transmission distant is 1200 meters. Because of the limitation of CP342-5 block the transmission rate is set at 1.5Mb/s, which is the most transmission rate of CP342-5.

Between master and slave the communication protocol is Profibus-DP master-slave protocol. The master reads data from slave and writes data to three slaves timely. To realize communication correctly enough I/O reflect section should be defined in master for every slave according to the data being exchanged in application. Because CP342-5 block is extended in every slave, the communication function is not integrated in the CPU block. The DP-SEND and DP-RECEIVE instructions should be used in the main program in every slave in order to realize data exchange between master and slave via CP342-5 block. DP-SEND and DP-RECEIVE instructions are used to send data in CPU to CP342-5 block and to receive data for CPU from CP342-5 block.

The RS-232c serial communication way between host computer and PLCs is often used in monitoring system. Though it is easy and simple, the transmission rate is low and transmission error is often occur. To communicate correctly the data check must be taken into account. In the NCS the communication way is PROFIBUS-DP bus between host computer and PLC master via CP5611 block, so communication veracity, in real-time and speediness are ensured. In host Visual Basic or Visual C programming language often designs computer the monitoring program.

How to design communication program to realize data communication in real-time between host computer and PLCs via Visual Basic or Visual C programming language? This is key problem in development of network control system. The paper presents the program design method to realize the data communication in detail and some communication programs in Visual Basic and Visual C used in Geotextiles-Laying Vesse NCS are offered.

3. PRODAVE S7 SOFTWARE

PRODAVE S7 software offered by SIEMENS Corp. is a kind of tool by which monitor computer can communicate with PLC. There are many functions based on Windows NT, Windows 95/98, Windows 3.11 or MS-DOS operation system in DLL of PRODAVE S7. Software engineers can use them in data communication between monitor computer and PLC in virtue of interface module, such as CP5411, CP5511, CP5611 or PC/MPI cable. It is not needed to engineers who must know some knowledge about communication and fieldbus well. Not only you can read data from PLC, but also can write data to PLC by using functions in PRODAVE S7 software under the Windows NT, Windows 95/98, Windows 3.11 or MS-DOS operation system. PRODAVE S7 software consists of two parts. One part is driver based on the Windows 95, Windows NT, Windows 3.11 and MS-DOS operation system. The other part is adapter of computer high language.

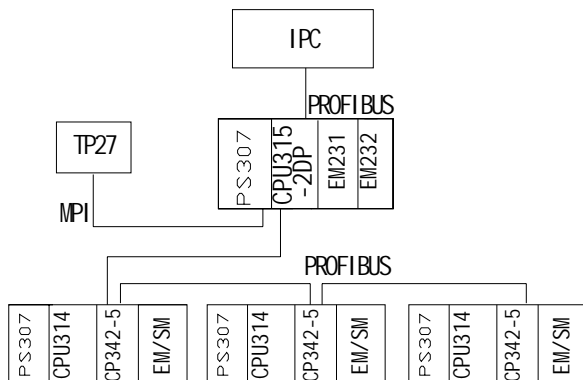


Fig.1 The system's structure chart

In this paper the functions in PRODAVE S7 software that we will use in communication between monitor computer and PLC are listed as follow:

- 1) **Load_tool** to establish the connection between monitor computer and PLC. When we call it there are three parameters being transferred, which is connection serial port number, the connection mode and the devices' address. The devices' address is a structure variable in that PLC's hardware information is encapsulation. A long integer is returned from this function. Zero returned from this function indicates the success of connection. For example: `res = load_tool (1, "S7ONLINE", plcadr)` means to establish NO.1 port connection in MPI mode with PLC which address parameters are indicated by "plcadr" variable.
- 2) **New_ss** to enable the connection established. For example: `"New_ss (1)"` means to enable the NO.1 connection.

- 3) **db_read** to read data in word from PLC DB block. There are four parameters needed to transfer in this function, which are the number of the DB block, the first address of the DB block, the quantity needed to read and data register address. Zero returned from this function indicates the success in reading data from PLC db block. For example `"res = db_read(10, 0, 5, value_word(0))"` means to read five data from DBW0 TO DBW4 in PLC DB10 block and store them to registers from `value_word(0)` to `value_word(4)`.
- 4) **Db_write** to write data to PLC DB block in word. The function format is the same with **db_read**. For example `"res = db_write(5, 2, 2, value_word(0))"` means to write the data in registers from `value_word(0)` to `value_word(1)` to DBW2 and DBW3 in PLC DB block.
- 5) **d_field_read** to read data from PLC in byte.
- 6) **d_field_write** to write data to PLC in byte.
- 7) **Unload_tool** to cut-off the connection between the monitor computer and PLC.

There are many other functions by which we can read data from PLC M block and V block or write data to there. By using these function we can realize many complex functions in reading and writing date.

4. THE COMMUNICATION BETWEEN MONITOR COMPUTER AND PLC

Visual Basic and Visual C are also the stronger visual programming languages and usually be used as a kind of development tool in designing monitor and control software. So how to design communication program based on Visual Basic and Visual C programming languages is important. We can call DLL or API functions in WIN32 to realize complex functions through Visual Basic or Visual C program.

4.1 Communication Program

Communication programs in Visual Basic based on PRODAVE s7 software package are listed as follow.

1) Function Declaration And Structure Variable Definition

To add function declaration in the module.

Declare Function `load_tool` Lib "w95_s7.dll" (ByVal nr As Byte, ByVal dev As String, adr As plcadrtype) As Long

Declare Function `new_ss` Lib "w95_s7.dll" (ByVal nr As Byte) As Long

Declare Function `unload_tool` Lib "w95_s7.dll" () As Long

Declare Function `db_read` Lib "w95_s7.dll" (ByVal db As Long, ByVal dw As Long, anz As Long, value As Integer) As Long

Declare Function `db_write` Lib "w95_s7.dll" (ByVal db As Long, ByVal dw As Long, anz As Long, value As Integer) As Long

2) Variable Definition

To add structure variable definition in the module.

Type `plcadrtype` 'to definite structure variable

adr As Byte 'PLC's address

SEGMENTID As Byte 'default is zero

SLOTNO As Byte 'the number of CPU's slot

RACKNO As Byte 'the number of rack

End Type

Public `plcadr` As plcadrtype 'to definite plcadr as a structural type

- 3) To Initialize and Establish Connection between Monitor Computer and PLC

To add codes into Form_load as follow:

```
plcadr.adr = 2      'PLC device's address is 2,
plcadr.SEGMENTID = 0 'the number of segment
is zero
plcadr.RACKNO = 0   'the number of rack is zero
plcadr.SLOTNO = 2    'the number of slot is 2
res = load_tool (1, "S7ONLINE", plcadr) 'If res
is zero, the connection is successful
If res <> 0 Then res = error_message (res, ErrorText)
'If res is not zero, the connection is unsuccessful.
```

- 4) To read data from PLC

```
Sub ReadDB ( BLOCKNO as long, no as long,
AMOUNT as long, Write_word(0) as value )
Dim I as integer
res=db_read (BLOCKNO, no, AMOUNT,
Write_word(0))
if res=0 then
for I=0 to amount-1
Return Value (i) =Write_word(i)
Next I
End if
End Sub
```

Communication programs in Visual C based on Prosave s7 software package are listed as follow:

- 1) To definite variable type

```
typedef struct
{
    UI adr ;           // host PLC's address
    UI SEGMENTID;      // host PLC network
    node address
    UI SLOTNO;         // the number of host PLC slot
    UI RACKNO;         // the number of host PLC rack
} plcadrtype;         //to definite plcadrtype type
```

- 2) To connect PLC

```
plcadr.adr=1
plcadr.SEGMENTID = 0
plcadr.RACKNO = 0
plcadr.SLOTNO = 2
res = load_tool(1, "S7ONLINE", plcadr)
if (res <> 0)
    MessageBox ("it is unsuccessful to connect PLC !")
else
    MessageBox ("it is successful to connect PLC, you
can continue")
```

- 3) To read data from PLC

when reading data from PLC timing, the program is as follow:

```
SetTimer(1,1000,NULL);
int i;
int BLOCKNO =1
int no =0
int AMOUNT = 1
res = new_ss(1) ;
if (res = 0)
res = d_field_read (BLOCKNO, no, AMOUNT,
value_byte(0)); // to read a data in type from host PLC
if (res <> 0) MessageBox ("error in reading
data!");
```

- 4) to cut-off the connection with PLC

When you exit application program, you should exit the connection.

```
res = unload_tool();
MessageBox ("disconnection with PLC ! ");
```

4.2 Data Communication Program Design

Data communication's initiative is finished in the monitor computer. PLC cannot request to send or receive data actively. When will the data be transferred is decided by monitor computer completely. Commonly in monitor program a timer is set in order to send or receive data timing. When the data quantity transmitted is bigger it will take long time to read or write data form monitor computer or PLC. So it will reduce system's reliability. Because the data in PLC will not vary in a scan period we could read or write PLC's data in time not timing to reduce monitor computer task. In this mode a special data storage block is defined, for example DB10 block. In it the data will be transferred are saved. DBW0 is defined as flag word and the other words after DBW2 in DB10 are used to save the data being transferred between monitor computer and PLC. Monitor computer detects flag word DBW0 constantly. Monitor computer checks the flag word timing. When flag word DBW0 is equal to FFFFH, monitor computer calls data transfer routine by which data are read from DB10 block. After data are read completely and processed in time, the flag word is reset to 0000H. The program flow chart is show in Fig.2. In this mode we must pay attention to modify DBW0's status flag word in time in PLC program. When the data being transferred is varied flag word DBW0 should be set in FFFFH immediately. It is easy to realize in PLC program..

5. CONCLUSION

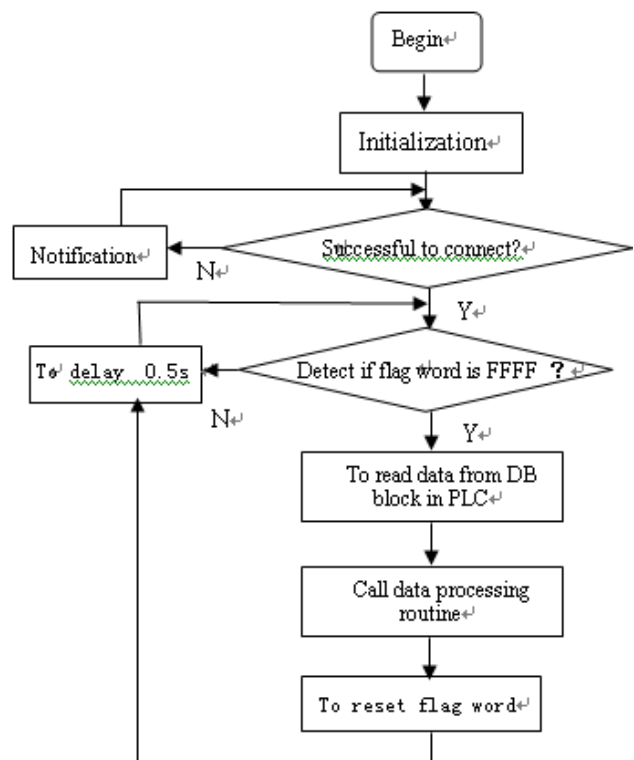


Fig.2 Program flow chart

In order to prevent data from being lost or machine died, there are three aspects to notice in designing data communication program in sum. First, before to end the program we must call unload tool function in order to disconnect PLC from monitor computer. Otherwise the monitor computer will be failure. Second, when monitor computer reads data from PLC the data transfer block must

be big enough because the adapter will not check-out data block. Third, the error file (ERROR.DAT) must be placed in the same subdirectory with monitor program or the adapter cannot find ERROR.DAT file.

The communication method between monitor computer and PLC introduced above is easy to master well. It is used in Geotextiles-Laying Vessel process control system successfully, the project is supported by ministry communications. The application result shows that error code rate is lower and data transfer efficiency is higher than conventional RS-232c serial communication method. And also data transfer rate is raised greatly.

6. REFERENCE

- [1] Tovar E, Vasques F. Cycle Time Properties Of The PROFIBUS Timed-Toked Protocol[J]. Computer Communication, 1999(22), PP:1206~1216.
- [2] Bogatyre V A, Estimation of The Mean Waiting Time for Transmission over A Reserved Channel with Restricted Communication Failure[J]. Automatic Control And Computer Science. 1998,32(1). PP:67~70.
- [3] Liao Ting-chang , Fang Yan-jun. Communication Interlinkage of Control System Based on PROFIBUS Fieldbus. Control Engineering of China. VOL.9 ,NO.4 , 2002 , PP : 32~34, in Chinese
- [4] Liu Qing, Zhou Tie-liang. The Design of Low-voltage Power distribution Supervision and Control system Based-on Profibus-DP. Electric Automation. VOL.25, NO.5, 2003, PP:74~75. in Chinese.
- [5] Li Bing-yu, Xiao Yun-shi. Present Application And Development Tendency of Networked Control System. Detect and control system. VOL.21, NO.4, 2002, pp:1~4, in Chinese
- [6] Guo Jan-ming. The Study of Computer Supervise and Control System Based on Fieldbus and Network. (D) M.S. paper of Wuhan University Of Technology. 2003.



Liu Qing is a Professor and vice dean of School of Automation, Wuhan University of Technology. She received her PH.D in Ship and Ocean Structure Object's Design and Manufacture from Wuhan University of Technology in 2002. She received her M.S. in Electric Drive and Automation from Wuhan Transportation University in 1988 and received her B.S in ship and port electric automation from Wuhan

Institute of water transportation and engineering in 1985. Her researching interests in computer control system and intelligent control theory and technology.

Address: 1040 He Ping Road

Postcode: 430063

Tel: 86-27-86548149

Fax: 86-27-87859049

E-MAIL: qliu2000@mail.whut.edu.cn

A Secure QoS Multicast Routing Protocol

Yang Mingxi, Li Layuan

School of Computer Science, Wuhan University of Technology

Wuhan, Hubei, 430070 China

E-mail: yangmx@mail.whut.edu.cn

ABSTRACT

The paper presents a simple and secure QoS (Quality of Service) multicast routing protocol. The protocol can operate in an open and insecure network environment and on top of the unicast routing protocol. The secure QoS multicast protocol is designed for the following goals: (1) Add security as QoS multicast routing metrics. (2) The key distribution protocol is both secure and robust in any open network environment. (3) It can support group membership authentication. (4) It can construct multicast trees with security and QoS constraints. (5) It provides confidentiality and authentication for multicast data transmission. This protocol make multicast receiver be able to choose between various service classes, each with different level of confidentiality and authenticity in a QoS aware multicast communication system.

Keywords: security, QoS, multicast, routing, protocol

1. INTRODUCTION

There are more and more Internet applications put new demands of QoS, multicast and security on the underlying network. In a Quality of Service (QoS) aware communication system, a user is able to choose between various service classes, each with different reliability, predictability, and efficiency degree. And QoS multicast protocol make multicast receiver be able to choose between various service classes. Several QoS multicast routing algorithms have been proposed recently. Algorithm proposed by Li Layuan et al. [1], i.e. QGMRP, is a novel QoS multicast routing Algorithm, which can significantly reduce the overhead for constructing a multicast tree with QoS constraints, but it is not secure. Some secure multicast protocols have also been provided, which mainly focus on the key management and secure group membership management. Algorithm proposed by Chu et al. in [2] present a secure multicast protocol with copyright protection, which built a secure and robust key distribution protocol in the presence of long delay or membership message. Adrian Perrig's algorithm proposed an Efficient and Secure Source Authentication for Multicast with symmetric cryptography [3][4], but which can't support QoS constrains. And several other people present some schemes to add security to QoS architectures [5][6], which discuss some initial ideas on how QoS architectures can be extended with a security dimension, but they are irrelative to multicast.

As mentioned above, until now no secure QoS multicast routing protocols have been provided, which implies that multicast receiver have no chance of configuring their level of Secure Quality of Service, and security is not recognized as metrics in current QoS multicast routing architecture.

This paper presents a secure QoS multicast routing protocol based on Li's QGMRP [1] and Chu's secure multicast

protocol [2] too. The proposed protocol can operate in an open and insecure network environment and on top of the unicast routing protocol. Our secure QoS multicast protocol is designed for the following goals: (1) Add security as QoS multicast routing metrics. (2) The key distribution protocol is both secure and robust in any open network environment. (3) It can support group membership authentication. (4) It can construct multicast trees with security and QoS constraints. (5) It provides confidentiality and authentication for multicast data transmission. This paper make multicast receiver be able to choose between various service classes, each with different confidentiality and authenticity in a QoS aware multicast communication system.

In the following, section 2 describes the network model [1]. In section 3 the idea of introducing security as metrics in current QoS architecture is discussed. Section 4 presents the secure QoS multicast protocol. Section 5 concludes the paper. Section 6 proposes the future work.

2. NETWORK MODEL

A network is usually represented as a weighted digraph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of communication links connecting the nodes. Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes. Let $s \in V$ be source node of a multicast tree, and $M \subseteq V - \{s\}$ be a set of end nodes of the multicast tree.

We also use $T(s, M)$ to denote a multicast tree, which has the following relations [1]:

- 1) $delay(p(s, t)) = \sum_{e \in p(s, t)} delay(e) + \sum_{n \in p(s, t)} delay(n).$
- 2) $cost(T(s, M)) = \sum_{e \in T(s, M)} cost(e) + \sum_{n \in T(s, M)} cost(n).$
- 3) $bandwidth(p(s, t)) = \min\{bandwidth(e), e \in p(s, t)\}$
- 4) $delay-jitter(p(s, t)) = \sum_{e \in p(s, t)} delay - jitter(e) + \sum_{n \in p(s, t)} delay - jitter(n).$
- 5) $packet-loss(p(s, t)) = 1 - \prod_{n \in p(s, t)} (1 - packet-loss(n)).$

Where for any node $n \in V$, $p(s, t)$ denotes the path from source node s to end node t of $T(s, M)$.

The QoS multicast routing problem is to find the $T(s, M)$ which satisfies some QoS constraints:

- 1) Delay constraint: $delay(p(s, t)) \leq D$
- 2) Bandwidth constraint: $bandwidth(p(s, t)) \geq B$
- 3) Delay jitter constraint: $delay-jitter(p(s, t)) \leq J$
- 4) Packet loss constraint: $packet-loss(p(s, t)) \leq L$

Meanwhile, the $cost(T(s, M))$ should be minimum. Where D is the delay constraint, B is bandwidth constraint, J is delay

jitter constraint and L is packet loss constraint. In the above QoS constraints, the bandwidth is concave metric, the delay and delay jitter are additive metrics, and the packet loss is multiplicative metric.

3. ADDING SECURITY AS QOS METRICS

QoS routing is an important function for the integrated services networks. High-speed networks are expected to support different applications, with different QoS requirements. Current and future communications would involve many critical information flows such as grid computing, financial transactions and private audio and video messages, which will introduce the question of how secure these network are.

A lot of effort has gone into the study of QoS metrics, such as delay, delay jitter and packet loss, as mentioned above. But a little, if any, work has investigated security as a QoS metric and its effect on routing. It is important to note that security can be applied at different levels in the network ranging from the system level to the physical layer level. So, in this section, we study security as a QoS metric, which reflects the security status of links. Considering the overhead, the two simple arguments below is our suggested security metrics so as to provide different Level of confidentiality and different Level of integrity:

- Type-cipher: {Plaintext, CAESAR, symmetric, asymmetric}
- Key-length: {0, 56, 128, 168, 192, 256, 512, 1024, 2048}

Where we all know that the key length of AES includes 128, 192, 256, 3DES includes 168, DES includes 56, and RSA includes 512, 1024 and 2048.

Here the assumption is that the longer keys the better confidentiality [6].

The one of the secure QoS multicast routing problem is to find the $ST(s,M)$ which satisfies some security and QoS constraints 1)2)3)4) as mentioned as section 2:

5) Type of cipher constraint: $type-cipher(p(n,t)) = TC$

6) Key length constraint: $key-length(p(n,t)) = KL$

Where TC is the Type of cipher constraint, KL is the Key length constraint.

4. THE PROPOSED SECURE QOS MULTICAST ROUTING PROTOCOL

4.1 Key Distribution Scheme

In our Secure QoS Multicast Routing Protocol (SQMRT), our key distribution algorithm requires only a *group leader* to be started. The *group leader* has the authority and the necessary information to accept/reject new membership request. The *group leader* may be given an access control list(ACL) which it can check if a new member can join it, or it can accept and verify a payment information as a mean for a new member to be admitted into the multicast group. We also assume that the address of the *group leader* is known to anyone who is interested to join the secure multicast group[6].

4.2 Group Membership Authentication

No matter who want to join a secure multicast group, sender or receiver, a requesting member first sends a *JOIN* request to the *group leader* using a secure unicast channel. The *group leader* checks its ACL to decide to either accept or reject the *JOIN* request. The *JOIN* contains the identity of the requesting member. We assume that both the requesting member and the group leader can properly authenticate each other in the secure unicast channel (e.g., via SSL) by presenting their certificates issued by well-known CAs (certificate authority) who authenticate their public keys. If the *JOIN* request is accepted, the group leader generates a unique member id *uid* that identifies the requesting member and then communicated the *uid* to the requesting member. Besides, the multicast session needs to establish a symmetric key K_{uid} between the *group leader* and the requesting member *uid* for later use in the multicast session. Generally if the secure unicast channel were setup to use RSA, the group leader would need to generate a symmetric key K_{uid} for the requesting member and transmits it to the new member. The member can now begin to join in the secure QoS multicast group and QoS multicast session according to the detail algorithm described in the following subsection.

4.3 Forming the Secure Searching Tree

In SQMRT, we add the secure metrics to the QoS constrains based on the algorithm [1][2].

The searching tree is formed incrementally. When a receiver g is a new member that wishes to join a multicast group G . The g sends *JOIN-G* and *rqst-g* to source of the group. We suppose the source of the group is also the *group leader* and u is an intermediate node of the path from g to s . When u receives the *rqst* message from the interface k , and V is a immediate upstream node of u . The s authenticate the g 's membership and u mainly check the security and QoS constrains below and decide its behavior:

```

if ( a new member  $g$  wants to join the multicast group)
   $g$  send JOIN-G upstream towards  $s$  by present it's
  certificate via secure unicast channel;
   $g$  and  $s$  authenticate each other;
  if not pass the authentication
     $g$  rev the REJECT from  $s$ ;
    exit;
  else
     $g$  rcv the ACCEPT AND uid from  $s$ ;
     $g$  send rqst-g upstream towards  $s$ ;
     $u$  rcv rqst from  $k$ ;
    if ( $d(g,k) + d(k,u) > D$ )
      ( $dj(g,k) + dj(k,u) > J$ )
      ( $bw(g,k) < B$ )
      ( $bw(k,u) < B$ )
      ( $tc(g,k) = TC$ )
      ( $tc(k,u) = TC$ )
      ( $kl(g,k) < KL$ )
      ( $kl(k,u) < KL$ )
      {  $u$  may satisfy the QoS constrains }
      for ( $RE(in,ou,m)$ ),do
         $RE.in = k$ ;
         $RE.out = \{V\}$ ;
         $RE.m = UR$ ;
        forward rqst to  $V$  ;
        skip
    fi
    if ( $d(g,f) + d(k,u) > D$ )

```

```

(dj(g,k)+dj(k,u)>J)
(bw(g,k)<B)  (bw(k,u)<B)
(tc(g,k)≠TC  tc(k,u)≠TC)
(kl(g,k)<KL  kl(k,u)<KL)
    transfer rjct to k;
fi
if (RE.m)=UR and rply is received
    for (FR(G,s,in,out,q)) do
        FR.in:=k;
        FR.out:={RE.in};
        FR.q:=rply.q;
        forward rply to RE.in;
        for each RE(in*,out*,m*) on the unicast routing
path
            m*=UR do
                FR.out:=FR.out+(UR*.in*)
                skip
            fi
        if (d(g,k)+d(*,s) > D)
            (dj(g,k)+dj(*,s) > J)
            (bw(i,j) < B)
            (tc(i,j)=TC)
            (kl(i,j) < KL)
            {all nodes on the unicast routing path satisfy the QoS
constrains}
            rply is sent back to g ;
            skip;
        fi
    if (d(g,k)+d(*,s)>D)
        (dj(g,k)+dj(*,s)>J)
        (bw(i,j)<B))
        (tc(i,j)≠TC)
        (kl(i,j)<KL)
        rjct is received
        RE.m*:=FR;
        RE.m:=FR;
    if (RE.m:=FR AND rjct is received)
        forward rqst to other neighbor nodes;
        {toward direction of s, but except for u}
        for (FR(G,s,in,out,q)),do
            FR.in := k
            FR.out: = {RE.in};
            FR.q:= rply.q;
            forward rply to RE.in;
            for each RE*(in*,out*,m*) on the fork routing paths
                m*=FR DO
                    FR.out:= FR.out+{RE*.in*}
            skip
        fi
        for (some possible and feasible paths from u to s) do
            if (d(g,k)+d(*,s) > D)
                (dj(g,k)+dj(*,s) > J)
                (bw(i,j) < B)
                (tc(i,j)=TC)
                (kl(i,j) < KL)
            rply, are sent back to g
            rcv rplys from the other neighbor nodes
        if (d(g,k)+d(*,s) > D)
            (dj(g,k)+dj(*,s) > J)
            (bw(i,j) < B)
            (tc(i,j)=TC)
            (kl(i,j) < KL)

```

```

(cost(p(s,g))=min[cost1, ..., costij]
    g rcv rply from a path;
    { which satifies the QoS constraints and has
minimum cost}
    skip;
    fi
fi
if (d(g,k)+d(*,s)>D)
    (dj(g,k)+dj(*,s)>J)
    (bw(i,j)<B))
    (tc(i,j)≠TC)
    (kl(i,j)<KL)
    rjct is received
    fi
fi

```

Where $d(*,s)$ and $dj(*,s)$ are the delay sum and the delay jitter sum from some intermediate node, such as immediate upstream node(u) of k , to the source of s of a path, respectively. The $d(k,*)$ and $dj(k,*)$ are the delay sum and the delay jitter sum from k to some intermediate node(which may also include the source s) of a path, respectively. The $bw(i,j)$ denotes the bandwidth of the link from i to j , i and j are any two adjacent nodes of path from the new member to source or some intermediate node. The $tc(i,j)$ denotes the type of cipher used between i and j , and $kl(i,j)$ denotes the key length used between i and j , i and j are any two adjacent nodes of path from the new member to source or some intermediate node.

According to the constrains above, we know, in this protocol the security is added to the QoS constrains as metrics enforcedly, except for the $tc(i,j)=plaintext$ $kl(i,j)=0$.

4.4 Forming the Secure Multicast Tree

The multicast tree is formed incrementally also. When a receiver g intends to join a multicast session, it sends a *JOIN-S* message to the *group leader* and *rqst-s* message to the source of the session, and initiates a routing process according to UR mode. When the group leader receive *JOIN-S* message and check the g 's *uid*, if it is true, a symmetric key K_{uid} is issued to g , the K_{uid} will be used between the *group leader* and the requesting member *uid* for later user in the multicast session. And an intermediate node (such as u) receives the *rqst-s* message, and it is already the on-tree node, it will make an eligibility test. It checks whether or not the QoS requirement of the new member as well as the existing QoS guarantees to the other on-tree members, can be fulfilled [1][2].

Suppose k is the immediate upstream node of g , the path delay and delay jitter from g to k are $d(g,k)$ and $dj(g,k)$, respectively. Similarly, the path delay and delay jitter from k to u should also be $d(k,u)$ and $dj(k,u)$, respectively. Let $bw(g,k)$ be the bandwidth of link from g to k and $bw(k,u)$ be the bandwidth of link from k to u . Recall that the delay, delay jitter and bandwidth constrains are D , J and B , respectively. The u will check

```

if (d(g,k)+d(k,u) > D)
    (dj(g,k)+dj(k,u) > J)
    (bw(g,k) < B)
    (bw(k,u) < B)
    (tc(g,k)=TC)
    (tc(k,u)=TC)
    (kl(g,k) < KL)
    (kl(k,u) < KL)

```

then it will continuously forward the *rqst* to its immediate upstream node. This process will be repeated until the source *s* receives the *rqst*. The *s* will check

```
if (d(s,*)+d(k,g) > D)
    (dj(s,*) +dj (k, g) > J)
    (bw(i,j) < B)
    (tc(i,j)=TC)
    (kl(i,j) < KL)
```

then it will send a *rply* message to the new receiver.

If the following equation holds

```
(d(g,f)+d(k,u)>D)
(dj(g,k)+dj(k,u)>J)
(bw(g,k)<B)
(bw(k,u)<B)
(tc(g,k)≠TC)
(tc(k,u) ≠TC)
(kl(g,k)<KL)
(kl(k,u)<KL)
```

then *u* will transfer a *rjct* message to its immediate downstream node *k*. At the moment, *k* enters the *FR* state.

Under the *FR* state, several paths will be searched. For these paths the *k* will check

```
if
(d(s,*)+d(k,g) > D)
(dj(s,*) +dj (k, g) > J)
(bw(i,j) < B)
(tc(i,j)=TC)
(kl(i,j) < KL)
```

then these paths are feasible paths. Among these feasible paths, the *k* will further check

```
if
(d(s,*)+d(k,g) > D)
(dj(s,*) +dj (k, g) > J)
(bw(i,j) < B)
(tc(i,j)=TC)
(kl(i,j) < KL)
(cost(p(s,g))=min[cost1,...,costij]
```

then the path that satisfy the above equation is just a optimal (or near-optimal) path for connecting the new member to the session. In the above *FR* routing searching process, the acceptance replies are sent the fork routing node *k*, the verification of the feasible paths and the selection of the optimal (or near-optimal) path are based on the accumulated metrics carried by reply messages.

Here, it is clear that the security is added to the QoS constrains as metrics enforcedly in the procedure of forming the multicast tree, except for the $tc(i,j)=plaintext$ $kl(i,j)=0$. Similar to the procedure of forming the searching tree. It is benefit to reduce the overhead and stronger the security.

4.5 Data Transmission

Data transmission can be divided into three phases: 1) the sending phase, 2) the verification phase, 3) the receiving phase[2]. The details of the three phase is simply described below.

- 1) the sending phase: when the sender multicasts an encrypted data message, the sender constructing a data message that contains 3 components:

$\{(suid, msgid), \{suid, msgid, data\}_{K_{msg}},$

$\{K_{msg}\}_{K_{suid}}\}$

where, *suid*—sender's member id

msgid—message id;

//the (*suid, msgid*) pairs uniquely identify a message in the multicast session.

K_{msg} — the key used to encrypt(via symmetric encryption) the user data,

// it is used only once for the current message, and a new key is randomly generated by the sender for the next message.

K_{suid} —the key established between the sending group member and the group leader when the sending //group member joins the multicast session, see section 4.4.

- 2) The verification phase: when the group leader multicasts a verification message that contains the key for //decrypting the data message.
- 3) The receiving phase: when the receivers receive both the data and verification messages and decrypt the data.

The details of this section please see Chu's algorithm [2].

4.6 Dynamic Membership Operations

We consider this section of our algorithm can reference to the section 3.2 of [2].

5. CONCLUSION

The multicast routing with QoS constrains is a very important service needed by a lot of multimedia application in Internet. And security is a much more important attribute of modern IT system. However, almost few work has been down about to combine security into QoS multicast routing protocol. This paper has try to present a protocol which support secure multicast routing with QoS constrains and security constrains.

This protocol make multicast receiver be able to choose between various service classes, each with different level of confidentiality and authenticity in a QoS aware multicast communication system.

In the future, We will continue to research the implementation or simulation of the SQMRT and the computing complexity.

6. ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under Grant No.60172035, 90304018 and NSF of Hubei Province.

7. REFERENCES

- [1] Li Layuan and Li Chunlin, "A new QoS multicast routing protocol", Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003. Proceedings of the Fourth International Conference, 27-29 Aug. 2003, pp: 32- 36.
- [2] Haohua Chu, Lintian Qiao and Klara Nahrstedt, "A Secure Multicast Protocol with Copyright Protection", ACM SIGCOMM Computer Communications Review, Volume 32, Number 2: April 2002, pp: 42-60
- [3] Adrian Perrig, Ran Canetti, Dawn Song, and J. D. Tygar. "Efficient and Secure Source Authentication for

- Multicast”, In Proceedings of the 2001 Symposium on Network and Distributed Systems Security (NDSS '01), February 2001
- [4] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. “Efficient Authentication and Signing of Multicast Streams over Lossy Channels”. In IEEE Symposium on Security and Privacy, May 2000.
 - [5] Syropoulou, Evdoxia, Agar, Christopher, Levin, Timothy, and Irvine, Cynthia, “ IPsec Modulation for Quality of Security Service”, Proceedings of the International System Security Engineering Association Conference, Orlando Florida, 13 March 2002
 - [6] Stefan Lindskog and Erland Jonsson. “Adding Security to Quality of Service Architectures”. In Proceedings of the SSGRR 2002s conference, L'Aquila, Italy, July 29-August 4, 2002
 - [7] Li Layuan and Li Chunlin, “A Multicast Routing Protocol with Multiple QoS Constraints”, <http://www.ifip.tu-graz.ac.at/TC6/events/WCC/WCC2002/papers/Layuan.pdf>



Yang Mingxi is a associate professor in School of Computer Science and Technology, Wuhan University of Technology in China. She graduated from Huazhong University of Science and Technology in 1982 with specialty of Automation. And now she is also a candidate of doctor in Wuhan University of Technology with specialty of

Computer Science and technology. Her research interests are in computer network and network security.



Li Layuan was born in Hubei, China on 26 February, 1946. He received the BE degree in Communication Engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in Communication and Electrical Systems from Huazhong University of Science and Technology, China in 1982. Since 1982, he has been with the Wuhan Wuhan University of

Technology in China, where he is a Professor, Advisor of Ph.D. and Editor in Chief of the Journal of WHUT (Edition of Transportation Science & Technology). He is Director of International Association for High Technology, Member of China Computer Federation and Membership of SPIE (USA) since 1986. He is also Paper Reviewer of IEEE INFOCOM, ICCS and ISRSDC. His research interests include high-speed computer networks, protocol engineering and image processing. Professor Li has published over 160 technical papers and is the author of seven books. He was awarded the National Special Prize by the Chinese Government in 1993.

Research of Indirect Replication Algorithm in Distributed Storage System*

Wang Yijie, Li Sikun

Institute of Computer, National University of Defense Technology, Changsha, Hunan, 410073, China

Email: wwwwyj1971@sina.com Tel.: (0731)4573663

ABSTRACT

Replication is the key technology of distributed storage system. In this paper, according to the intrinsic characteristic of distributed storage system, based on the peer-to-peer model, the indirect replication algorithm is proposed. In the indirect replication algorithm, the data object is partitioned into several data blocks, and then these data blocks are encoded in order that there is data redundancy between data blocks. Compared with the traditional replication algorithm, the indirect replication algorithm has less granularity of replication, less bandwidth cost and storage cost, and can provide higher availability, durability and security. The results of performance evaluation show that the encode time and decode time is proportional to the square of data size, and that if the number of encoded data blocks used to recover data object increases, the decode time is decreased greatly.

Keywords: Distributed storage system, replication, data availability, performance evaluation.

1. INTRODUCTION

Peer-to-peer computing has emerged as a significant social and technical phenomenon. In the peer-to-peer model, many peers work together in a symmetrical way to provide storage services, clients need not install any new software to consume basic storage services. The distributivity, autonomy, dynamic, scalability and flexibility of peer-to-peer model can be utilized to improve the efficiency of distributed heterogeneous storage resources. Therefore, many research projects on distributed storage are based on peer-to-peer model, for example, Freenet [1], Oceanstore [2], Chord [3] and StarFish [4].

The traditional replication technology can be used to achieve the availability and durability of distributed storage system [5,6,7,8,9]. In the traditional replication technology, the replica size is equal to data size; in order to guarantee the availability and durability, the number of replicas need to be increased while the system size increased; however, to some extent, more replicas need more network bandwidth and more storage capacity, especially for large data object; on the other hand, the increasing replicas also influence data security. Therefore, the traditional replication technology is not adequate for data management in distributed storage system very well.

In this paper, according to the intrinsic characteristic of distributed storage system, based on the peer-to-peer model, the indirect replication algorithm is proposed. Compared with the traditional replication algorithms, the indirect replication algorithm has less granularity of replication, less bandwidth

cost and storage cost, and can provide higher availability, durability and security. Section 2 describes the indirect replication algorithm in detail. Section 3 presents the results of performance evaluation. In Section 4 we draw conclusions from our research work so far.

2. INDIRECT REPLICATION ALGORITHM

In the indirect replication algorithm, data object D is partitioned into m data blocks d_1, d_2, \dots, d_m which are of the same size b , then the m data blocks are encoded into n encoded data blocks of size b by Reed - Solomon code [10,11], there is data redundancy between data blocks, so that the data replication is realized indirectly, the n encoded data blocks are distributed among several storage nodes; if some read requests about data object D are received, any m encoded data blocks of n encoded data blocks can recover the data object D . $e_r = m/n$ named as code rate. Reed - Solomon code is systematic code, so the first m data blocks of n encoded data blocks are the same as the m data blocks of data object D , which are named as information blocks, the other $n-m$ data blocks of n encoded data blocks are named as check blocks.

The key components of indirect algorithm are encoding and decoding.

The generator matrix G is utilized to encode the data blocks d_1, d_2, \dots, d_m into n encoded data blocks $d_1, d_2, \dots, d_m, c_1, c_2, \dots, c_{n-m}$, that is to say, $D \bullet G = [DC]$, of which, $C = [c_1, c_2, \dots, c_{n-m}]$, $G = [I_m F]$, I_m is an identity matrix. Therefore, $D \bullet F = C$. We define F to be the $m \times (n-m)$ Vandermonde matrix, so $D \bullet F = C$ becomes:

$$\begin{bmatrix} d_1 & d_2 & \dots & d_m \end{bmatrix} \times \begin{bmatrix} 1 & 1 & \Lambda & 1 \\ x_1 & x_2 & \Lambda & x_{n-m} \\ x_1^2 & x_2^2 & \Lambda & x_{n-m}^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^{m-1} & x_2^{m-1} & \dots & x_{n-m}^{m-1} \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & \dots & c_{n-m} \end{bmatrix} \quad (1)$$

x_1, x_2, \dots, x_{n-m} are all nonzero, and they are different from each other.

$D \bullet G = [DC]$ becomes:

$$\begin{bmatrix} d_1 & d_2 & \dots & d_m \end{bmatrix} \times$$

* This work is supported by the National Grand Fundamental Research 973 Program of China (No.2002CB312105), A Foundation for the Author of National Excellent Doctoral Dissertation of PR China (No.200141), and the National Natural Science Foundation of China (No.69903011, No.69933030).

$$\begin{bmatrix} 1 & 0 & \Lambda & 0 & 1 & 1 & \Lambda & 1 \\ 0 & 1 & \Lambda & 0 & x_1 & x_2 & \Lambda & x_{n-m} \\ 0 & 0 & \Lambda & 0 & x_1^2 & x_2^2 & \Lambda & x_{n-m}^2 \\ M & M & M & M & M & M & M & M \\ 0 & 0 & \Lambda & 1 & x_1^{m-1} & x_2^{m-1} & \Lambda & x_{n-m}^{m-1} \end{bmatrix}$$

$$= [d_1 \ d_2 \ \dots \ d_m \ c_1 \ c_2 \ \dots \ c_{n-m}] \quad (2)$$

We define matrix N to be $[DC]$, it will be seen from formula (2) that if m data blocks are selected from n encoded data blocks, the corresponding rows of the matrix G and N are kept, and the other rows are deleted. Therefore, we get matrix G' and N' , and $D \bullet G' = N'$. Because the matrix F is defined to be a Vandermonde matrix, every subset of rows of matrix G is guaranteed to be linearly independent. Thus, the matrix G' is non-singular, and the values of matrix D may be calculated from $D \bullet G' = N'$ using Gaussian Elimination. Hence all m data blocks of data object D can be recovered.

The computational complexity of indirect replication algorithm is $O((mb)^2)$.

3. PERFORMANCE EVALUATION

In this section, the performance of indirect replication algorithm is analyzed. Firstly, the indirect replication algorithm is compared with the traditional replication algorithms; secondly, the influence of data size on the efficiency of indirect replication algorithm is analyzed; lastly, the relation between the number of data blocks used to recover the data object and the decode time of indirect replication algorithm is studied.

3.1 Comparative Analysis of Indirect Replication Algorithm and the Traditional replication algorithms

The comparative analysis of indirect replication algorithm and the traditional replication algorithms involves data availability, storage cost, and bandwidth cost.

◆ Data Availability

In the indirect replication algorithm, the data object D is partitioned into m data blocks, the m data blocks are encoded into n encoded data blocks, any m data blocks of n encoded data blocks can be decoded into m data blocks of D . There are N nodes in the distributed storage system, M nodes of which are unavailable. Then, the availability of data object D can be expressed as

$$P_D = \sum_{i=0}^{n-m} \left(\binom{M}{i} \binom{N-M}{n-i} \right) / \binom{N}{n}.$$

It is assumed that there are 10000 nodes in the distributed storage system, 1000 nodes of which are unavailable. For the traditional replication algorithms, if there are 2 replicas of data object D , then the data availability of D is 0.99; if there are 3 replicas of data object D , then the data availability of D is 0.999. For the indirect replication algorithm, the data object D is partitioned into 20 data blocks, if the 20 data blocks are encoded into 40 encoded data blocks, and any 20 encoded data blocks of which can be decoded into 20 data blocks of D , then

the availability of D is 0.999999998; if the 20 data blocks are encoded into 60 encoded data blocks, and any 20 encoded data blocks of which can be decoded into 20 data blocks of D , then the availability of D is $(1-10^{-25})$. It is observed that the indirect replication algorithm can get higher data availability than the traditional replication algorithms with the same storage cost and bandwidth cost.

◆ Storage Cost

In the traditional replication algorithms, the storage cost of data object D is $S_T = RD$, R is the number of replicas. In the indirect replication algorithm, the storage cost of data object D is $S_{DP} = D/e_r$, e_r is the code rate. Therefore, $S_T/S_{DP} = (RD)/(D/e_r) = R \bullet e_r$.

According to the analysis of data availability, if the traditional replication algorithms gain the same availability as the indirect replication algorithm, R should be greater than $1/e_r$, so $S_T/S_{DP} > 1$, which is to say that the storage cost of the traditional replication algorithms is greater than that of the indirect replication algorithm.

◆ Bandwidth Cost

The bandwidth cost of replication algorithm includes the bandwidth cost of read access and the bandwidth cost of write access.

In the traditional replication algorithms, the bandwidth cost of read access is the size of data object D , $W_{tr} = D$. In the indirect replication algorithm, D is partitioned into m data blocks, the m data blocks are encoded into n encoded data blocks, any m encoded data blocks of which can recover D , the bandwidth cost of read access is the size of encoded data blocks used to recover data object D , $W_{DP_r} = (m/m)D = D$. $W_{tr}/W_{DP_r} = 1$, which is to say that the bandwidth cost of read access of the traditional replication algorithms is equal to that of the indirect replication algorithm.

In the traditional replication algorithms, the bandwidth cost of write access is $W_{tw} = RD$, R is the number of replicas. In the indirect replication algorithm, the bandwidth cost of write access is $W_{DP_w} = D/e_r$, e_r is the code rate. Therefore, $W_{tr}/W_{DP_w} = RD/(D/e_r) = R \bullet e_r$. According to the analysis of data availability, if the traditional replication algorithms gain the same availability as the indirect replication algorithm, R should be greater than $1/e_r$, so $W_{tw}/W_{DP_w} > 1$, which is to say that the bandwidth cost of write access of the traditional replication algorithms is greater than that of the indirect replication algorithm.

3.2 Influence of Data Size on Efficiency of Indirect Replication Algorithm

In the indirect replication algorithm, the data object D is partitioned into m data blocks, the m data blocks are encoded into n encoded data blocks, any m encoded data blocks can recover D . The indirect replication algorithm is implemented in C, the experimental environment is P4 1.5G microcomputer with 256M Bytes memory. The size of data block is set to 1KB, the code rate is set to 0.5. We change the data size, observe the encode time and decode time (Table 1). The computational complexity of the indirect replication algorithm is $O((mb)^2)$, which is to say that the computational complexity is proportional to the square of data size. As showed in Table 1, the test result is consistent with the theoretical analysis on the whole.

Table 1 influence of data size on efficiency of indirect replication algorithm

Data Size (B)	Encode Time (s)	Decode Time (s)
250K	1.093	0.654
500K	4.330	2.610
1M	18.913	10.904
2M	81.098	42.025
4M	340.790	173.139
8M	1480.453	750.903

3.3 Relation between the Number of Encoded Data Blocks Used to Recover Data Object and the Decode Time of Indirect Replication Algorithm

In the indirect replication algorithm, the data object D is partitioned into m data blocks, the m data blocks are encoded into n encoded data blocks, any m encoded data blocks can recover D. If more encoded data blocks are used to recover D, the decode time maybe different. In this section, the relation between the number of encoded data blocks used to recover data object and the decode time of indirect replication algorithm is analyzed. In the following experiment, the size of data object is set to 500KB, the size of data block is set to 1KB, the code rate is set to 0.5. For each value of NUM, which is the number of encoded data blocks used to recover data object, the different data blocks are selected to recover data object for 1000 times, and the average of decode time is computed (Figure 1). In Figure 1, the decode time is decreased greatly as NUM increases. Because the complicate matrix operations maybe reduced as the number of encoded data blocks used to recover data object increases.

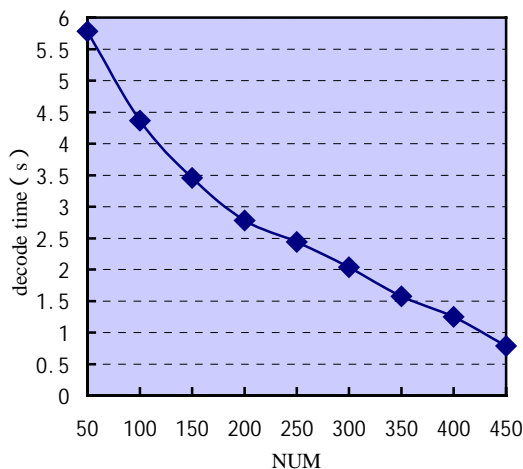


Figure 1 the relation between NUM and the decode time of indirect replication algorithm

4. CONCLUSION

In the indirect replication algorithm, firstly, the data object is partitioned into several data blocks; secondly, these data blocks are encoded in order that there is data redundancy between data blocks, so that the data replication is realized indirectly, the high availability and durability of distributed storage system are achieved, and the data security is guaranteed to a certain extent. The results of performance evaluation show that the encode time and decode time is proportional to the square of data size, and that if the number of encoded data blocks used to recover data object increases, the decode time is decreased greatly.

5. REFERENCES

1. Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore W. Hong, Freenet: A Distributed Anonymous Information Storage and Retrieval System, Lecture Notes in Computer Science, vol 2009:46-59, 2001.
2. John Kubiawicz, David Bindel, Yan Chen, Steven Czerwinski, et al., OceanStore: An Architecture for Global-Scale Persistent Storage, Proc. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS-IX), pp190-201, ACM Press, New York, 2000.
3. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, pp160-177, San Deigo, CA, August 2001.
4. Eran Gabber, Jeff Fellin, Michael Flaster, Fengrui Gu, et al., StarFish: highly-available block storage, 2003 USENIX Annual Technical Conference, pp151-163, San Antonio, TX, USA, June 9-14, 2003.
5. Edith Cohen, Scott Shenker, Replication strategies in unstructured peer-to-peer networks, in The ACM SIGCOMM'02 Conference, pp308-321, Pittsburgh, USA, August 2002.
6. Wang Yijie, Jiang Xueyang, Research of Replication Techniques in Internet Distributed Storage System, Journal of Computer Research and Development, 40(Suppl.):30-35, 2003.
7. C.G. Plaxton, R. Rajaraman and A.W. Richa, Accessing Nearby Copies of Replicated Objects in a Distributed Environment, In Proc. 9th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA '97), pp311-320, Newport, RI, USA, 1997.
8. Andrzej Duda, Analysis of Multicast-Based Object Replication Strategies in Distributed Systems, In Proceedings of the 13th International Conference on Distributed Computing Systems, Pittsburgh, USA, pp311-318, 1993.
9. Jussi Kangasharju, James Roberts, Keith W. Ross, Object Replication Strategies in Content Distribution Networks, In Proceedings of WCW'01: Web Caching and Content Distribution Workshop, pp252-261, Boston, USA, June 2001.
10. Johannes Blömer, Malik Kalfane, Richard Karp, Marek Karpinski, Michael Luby, David Zuckerman, An XOR-Based Erasure-Resilient Coding Scheme, Technical report, International Computer Science Institute, Berkeley, California, 1995.
11. Luigi Rizzo, Effective Erasure Codes for Reliable Computer Communication Protocols, Computer Communication Review, 27(2):24-36, April 1997.



Wang Yijie was born in 1971. She received her Ph.D. degree from National University of Defense Technology in 1998. Now she is an Associate Professor of National Key Laboratory For Parallel & Distributed Processing, Institute of Computer, National University of Defense Technology. Her research interests include network computing, database and mobile computing.

Wang Yijie was born in 1971. She received her Ph.D. degree from National University of Defense Technology in 1998. Now she is an Associate Professor of National Key Laboratory For Parallel & Distributed Processing, Institute of Computer, National University of Defense Technology. Her research interests include network computing, database and mobile computing.

Design and Performance Analysis of High Availability iSCSI Storage Area Network*

Jiang Minghua^{1,2} Zhou Jingli¹

¹ Key Laboratory of Data Storage System, Huazhong University of Science and Technology
Ministry of Education

Wuhan, Hubei, 430074, China

² Department of Computer Science & Technology, Wuhan University of Science & Engineering
Wuhan, Hubei, 430074, China

Email: mhjiang@126.com Tel.: 027-87802441.

ABSTRACT

iSCSI is emerging as an end-to-end protocol for transporting storage I/O block data over IP networks. By exploiting the ubiquitous Internet infrastructure, iSCSI greatly facilitates remote storage, remote backup, data mirroring and iSCSI-based SAN. This paper describes a design and implementation of high availability iSCSI SAN to improve performance by using distributed iSCSI RAID and improve availability by using failover storage servers, then this paper discusses Markov model and I/O performance analysis of this system.

Keywords iSCSI, Distributed iSCSI RAID, iSCSI SAN, Availability Model, Performance Analysis

1. INTRODUCTION

The rapid growth of data intensive applications, such as simulation, modeling, Internet browsing, multimedia, transaction processing, e-business, and data mining are continuously driving the demands for large amounts of data storage volumes as well as demands for fast and reliable access to the data. Storage Systems have evolved from traditional application-level file servers to large, independent network-aware, disk-array systems such as NAS[1] and SANs[2]. Storage Area Networks implement a networked storage model where a layer of switches connects the servers of the SAN to the storage devices of the SAN. SANs allow virtualization and sharing of devices, as well as facilitated backup and recovery. Traditionally SANs have been designed using Fiber Channel technology that overcomes the SCSI-3 limitations in terms of the maximum cable lengths, and can be extended up to 10km using standard single-mode fiber even though copper and multi-mode fiber is also supported for shorter distances. Recently, alternatives to Fiber Channel SANs based on IP technology have begun to appear. Arguably the most interesting of these is iSCSI, which encapsulates SCSI commands and data blocks into the TCP/IP protocol messages. The servers in an distributed iSCSI Storage system are called iSCSI initiators and the storage devices act as iSCSI targets. The iSCSI targets and initiators can be connected using standard Ethernet switches, meaning that a lower cost solution is possible since the per-port cost of Ethernet switches is currently much lower than the per-port cost of Fiber Channel switches. With iSCSI you can convert any IP fabric into a SAN, which is a network used to carry storage traffic. Meanwhile, Faced with increased customer and internal user

expectations, companies are currently striving to achieve the availability needed to support 24x7x365 uptime data access requirements by utilizing highly available components and solutions as well as a fault tolerant design.

RAID (Redundant Array of Independent Disks)[3] is a mature technique to improve performance and reliability of disk I/O through parallism and redundancy. This paper describes a high availability (HA) storage area network base on iSCSI technology, this HA iSCSI SAN is consisted of front-end storage service subsystem and backend Distributed iSCSI RAID subsystem that stripe data among several iSCSI targets. A fault tolerant of this system consists of three layers that include RAID fault-tolerant, redundant networking for robust connections and HA servers to failover. Software implemented fault-tolerance and system executive are used to provide fault-detection and error recovery.

The rest of the paper is organized as follows: section 2 presents the design and implementation of distributed iSCSI storage system. Section 3 presents availability Markov model of this system. Performance analysis is described in Section 4. The conclusions and potential future research are presented in Section 5.

2. DESIGN AND IMPLEMENTATION OF HA iSCSI SAN

2.1 HA iSCSI SAN Architecture

We introduce HA iSCSI SAN to solve the performance and high availability by a combination of well-managed hardware, software components and shared a pool of Distributed iSCSI RAID (Figure1).

Two storage servers are connected by redundant IP switch through redundant I/O channels to handle failover seamlessly. Only healthy storage servers are allowed to participate in the delivery of storage services for clients. The health of each individual sever, along with its hardware and software components, is active monitored. Failing or failed servers are prevented from delivering services and accessing data. Failed software components can be restarted within the system, and failed servers may return to the system following repair. Distributed iSCSI RAID is to organize the iSCSI targets similar to RAID by using rotated parity techniques, each iSCSI target is a basic storage unit in the array, and it server as storage device node. All the nodes in the array are connected to each other through a high-speed switch to form a local area network.

* This paper is supported by national natural science foundation of china (No. 60373088)

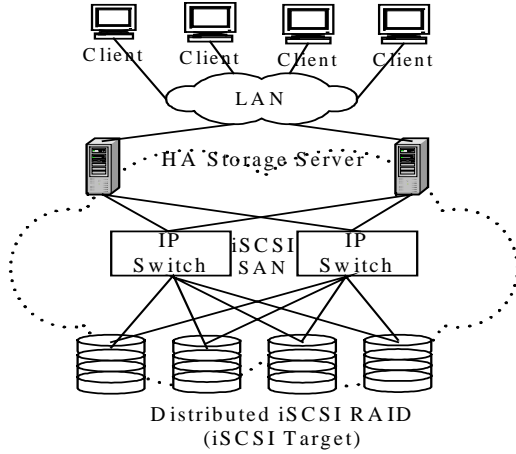


Figure 1 Architecture of High Availability iSCSI Storage Area Network

2.2. Data Placement of Distributed iSCSI RAID

To increase aggregate performance by ganging together multiple targets in parallel and to improve the reliability, Distributed iSCSI RAID interleaves data across multiple iSCSI targets. To protect against single iSCSI target failures, Distributed iSCSI RAID adds a parity block for each row of data blocks in different nodes. These parity blocks have the same size and are rotated between the storage device nodes as shown in Figure 2, where the shadowed blocks are parity blocks, and others are data blocks. Each parity block contains the parity for the data blocks in its row. For example $P_0 = D_{00} \oplus D_{10} \oplus D_{20}$.

iSCSI Target 0	iSCSI Target 1	iSCSI Target2	iSCSI Target 3
D ₀₀	D ₁₀	D ₂₀	P ₀
D ₀₁	D ₁₁	P ₁	D ₃₁
D ₀₂	P ₂	D ₂₂	D ₁₂
P ₃	D ₁₃	D ₂₃	D ₃₃
...

Figure 2 Data Placement of Distributed iSCSI RAID

2.3. HA Management

In order to guarantee data consistency and service availability, it is extremely important that the system as a whole agrees on membership-related storage servers. If primary server failed, the backup server can take over all resources belonging to the failed server to ensure continuous storage service such as E-mail, FTP, DBMS and WWW etc. HA management achieves these goals:

- **Fault Detection:** Fault detection are key issues in implementation of HA iSCSI SAN. The heartbeat is basically a key mechanism for fault detection by using IP protocol over two servers. Additionally, this system provides point-to-point serial connections as heartbeat channels. Two servers send periodic heartbeats message "I am alive" to each other to indicate a likely fault. If a message is not replied to within a specified time out, a

failover occur.

- **Services Infrastructure:** ability to define high availability services and provide a means of having them stopped and started in response to state transitions.
- **System Management:** allowing the specification of configuration and tuning parameters, as well as monitoring current operational status.
- **Service Takeover:** In the event of a server failure, the other server takes over all the services of the failed one in such a way as to minimize disruption to the network users. Service takeover is achieved by using IP and MAC address takeover from the failed server onto an unused Ethernet card on the takeover server.
- **Shared Storage:** Data sharing is one of the fundamental design goals for a HA iSCSI SAN. All iSCSI target attached to the switch are immediately visible to each server node when the node boots. Therefore, all files and RAID volumes created on these devices can be accessed directly by each server node. Once remounted, this new file system will be visible to two servers and all files and RAID volumes are attributed to "LOCAL" to each server.

3. AVAILABILITY ANALYSIS

The modeling methodology chosen for this evaluation was a state-space model. Assumptions were made to reduce the size of the state space and to focus on failure mechanisms of the storage service subsystem and distributed iSCSI RAID subsystem. We assume that the time to failure and the time to repair for each are exponentially distributed. We also assume that whenever the system is down, no further failure can take place. The model also excludes elements that are not considered for the purpose of providing availability guarantees. Denote the steady-state availability of the two subsystems as A_F and A_B respectively. The steady-state availability of the whole system can then be computed $A_{SYS} = A_F \cdot A_B$.

3.1 Availability Model of Storage Service Subsystem

The Markov model [8] of storage service subsystem is shown in Figure 3, (2,2) as the state where both two servers is up and two network link are available, (1,2) as state where only one server is up and two network link are available, (0,0) as state where no server is up and two network link are unavailable, and so on. Here,

λ_s : Failure rate of storage server; λ_w : Failure rate of network link;

μ : Repair rate of storage server; β : Repair rate of network link

The steady-state availability A_F is given by

$$A_F = \frac{1 + \frac{2\lambda_s}{\mu} + \frac{2\lambda_w}{\beta} + \frac{4\lambda_s\lambda_w}{\mu\beta}}{E},$$

Where

$$E = 1 + \frac{2\lambda_s}{\mu} + \frac{2\lambda_L}{\beta} + \frac{2\lambda_s^2}{\mu^2} + \frac{2\lambda_L^2}{\beta^2} + \frac{4\lambda_s\lambda_L}{\mu\beta} + \frac{4\lambda_s^2\lambda_L}{\mu^2\beta} + \frac{4\lambda_s\lambda_L^2}{\mu\beta^2}$$

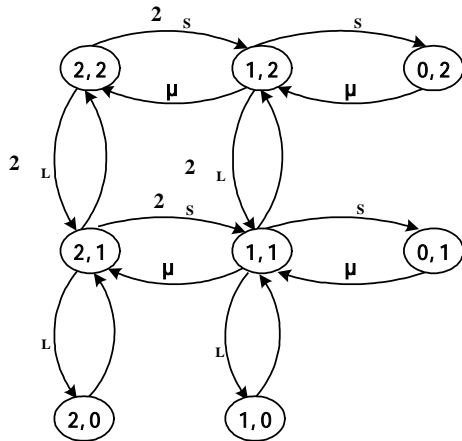


Figure 3 Markov Model of Front-end Storage Service Subsystem

3.2. Availability Model of Distributed iSCSI RAID Subsystem

Figure 4 is Markov mode of backend distributed iSCSI RAID subsystem, this subsystem uses

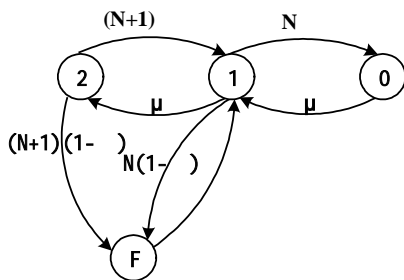


Figure 4 Markov Model of Backend Distributed iSCSI RAID Subsystem

$N+1$ ($N=15$) SCSI disk in RAID 5, In state 2, all the disks are operational. If a covered disk failure is correctly predicted with probability ρ , then system state does not change (assuming unlimited number of spares). A spare disk is switched in and data is reconstructed onto spare, then the subsystem state change from 2 to 1, if an uncovered failure occurs in state 2, then subsystem changes state from 2 to F. An uncovered disk failure while reconstruction is underway causes transition from state 1 to F. An unpredicted covered failure in state 1 causes a transition to state 0, data reconstruction rate in state 0 and state 1 is μ , where μ is the repair transition rate from the failure state F to state 1. The expression for the steady-state availability of the

subsystem is as follows[12]:

$$A_B = \frac{1 + \frac{2\lambda}{\mu}}{1 + \frac{2\lambda}{\mu} + \frac{2\lambda(1-\rho)}{\beta} + \frac{2\lambda^2}{\mu^2}}$$

4. PERFORMANCE ANALYSIS

For the purpose of performance analysis, we have implemented HA iSCSI SAN prototype based on Linux platform, test environment includes six PC servers (NL120), namely PCS1 through PCS6, PCS1-2 serve as fault-tolerance management and as iSCSI Initiators, PCS3-6 are four iSCSI targets, which are organized as distributed iSCSI RAID. The default data block size is set to 64kB. All PC servers are interconnected through two Cisco 3550-12T to form a HA iSCSI SAN. Each machine is running Linux kernel 2.4.18 integrating with a Intel 1000BaseT network interface card and Adaptec 7899 high performance SCSI adaptor, in addition, each PCS appends a Intel 1000BaseT NIC as a redundancy for high availability. The configurations of these machines are described in Table 1.

Table 1: The configurations of machines

Machines	HA Storage Server	iSCSI Target
Processor	Inter Xeon2.0	Inter Xeon2.0
RAM	1024MB	1024MB
SCSI Controller	Adaptec 7889	Adaptec 7889
SCSI Disk	Compq BD03695A	4 × Compq BD03695A

We use popular file system benchmark tool Iozone to measure the performance of sequential read/write, random read/write that be generally the primary concerns for any storage systems. The average of throughput listed here is the arithmetic average of above four I/O operations. Figure 5 shows the average throughputs. The data set is under 1G bytes and I/O request sizes range from 4kB to 64KB.

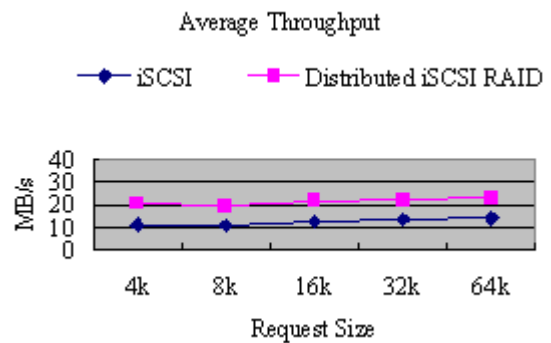


Figure 5 Distributed iSCSI RAID I/O Performance

5. CONCLUSION

iSCSI is a new emerging protocol with the goal of implementing the storage area network technology over TCP/IP which brings economy and convenience whereas it also raise performance issue. High availability systems have widely requirement in today's rapid increasing Internet service. We present a design and implementation of High availability storage system base on iSCSI, called it as HA iSCSI SAN. We describe the Markov model of this system. and analyze its I/O performance.

6. REFEEENCES

- [1] Gibson, G.A., R. Van Meter, "Network Attached Storage Architecture," Comm. of the ACM, Vol. 43, No 11, November, 2000, pp. 37-45.
- [2] Barry Phillips. "Have Storage Area Networks Come of Age? " IEEE Computer, Vol.31, No.7, 1998, pp. 10-12.
- [3] D. A. Patterson, G. A. Gibson, R. H. Katz, "A Case for Redundant Arrays of Inexpensive Disks (RAID)," Proceedings of the International Conference on Management of Data (SIGMOD), June 1988, pp. 109-116.
- [4] Marcos Kawazoe Aguilera, Wei Chen, and Sam Toueg. "Heartbeat: a timeout-free failure detector for quiescent reliable communication," Proceedings of the 11th International Workshop on Distributed Algorithms, Lecture Notes on Computer Science. Springer-Verlag, September 1997, pp. 126-140.
- [5] Yu Shengsheng, Liu Nian, and Zhou Jingli, "Design and Implementation of Storage Network Management System based on FC-SAN," Mini-Micro Systems, Vol. 23, No. 4, 2002, pp.401-404.
- [6] J.T. Blake and K.S. Trivedi, "Reliability analysis of interconnection networks using hierarchical composition," IEEE Transactions on Reliability, Vol. 38, No. 1, Apr. 1989, pp. 111-120.
- [7] Manish Malhotra and Kishor S. Trivedi. "Reliability modeling of disk array systems, " Sixth International Conference on Modeling Techniques and Tools for Computer Performance Evaluation, Edinburgh, Scotland, Sept. 1992, pp. 16-18.
- [8] KS Trivedi. Probability and Statistics, with Reliability, Queuing, and Computer Science Applications. Prentice Hall, Englewood Cliffs, NJ, 1982.

A High Performance Dynamic Memory Management Scheme

Huifu Zhang^{1,2} Fangmin Li^{1,2}

¹School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201

²School of Information engineering, Wuhan university of technology, Wuhan, 430070

Email: afu@sina.com Tel:008613554388001

ABSTRACT

Dynamic Memory Management takes an important count in most programs. Compared with other DMMs, first fit DMM using address-ordered linear list data structure is the most researched and applied, since it tends to enhance program locality, and causes significantly less fragmentation. In this paper, an efficient scheme suitable for this kind of DMM is proposed which can evidently improve the DMM efficiency, especial memory-releasing efficiency, by caching recently freed blocks. This improvement is testified by theoretical analysis and evaluation result.

Keywords: Dynamic Memory Management, CACHE, address-ordered, first fit

1. INTRODUCTION

Dynamic Memory Management (DMM) takes an important count in most programs. There are many researches show that DMM occupies 30% running time in C programs that operate the memory frequently. In object-oriented program design language C++, creating and deleting objects frequently makes the memory operation more frequently. In order to promote the running efficiency, high-performance DMM has been attached more importance. At present, special DMM algorithms are adopted in many large application systems. It furnishes DMM efficiency at one hand and another recovers rubbish memory. This paper takes executing efficiency in time of DMM into account.

The storage capacity whole DMM maintains may gives birth to many alternations between memory blocks in use and free ones because of the size of memory the application applies and the random of arriving time and TTL (time to live) of memory objects. Therefore, it is necessary to express the information of accessible memory blocks, adjacent blocks of DMM in certain ways so as to apply for or release the memory blocks.

In the past 40 years of theoretical research and practical system, DMM usually uses linear list data structure, which makes accessible memory blocks connected with a certain bi-direction list. The forms of list are FIFO, FILO and address-ordered. To FIFO list, DMM puts memory blocks released each time at the head of the list; To FILO list, DMM puts memory blocks released each time at the tail of the list; To the address-ordered list, DMM seeks the suitable inserting position to keep the sequence of the list when it release memory block into it. The algorithms of allocating memory blocks contain optimal fit, first fit, next fit and worst fit. The realization of direct linear list organizes all usable memory blocks (arbitrary size) in a linear list. Advantages of this are

that it allows memory blocks in arbitrary size and the combination of adjacent blocks is simple and efficient, which benefits to decrease fragmentations. But, flexibility of it is inferior. It will take long time to seek when the linear list is large. One modification is diff-free list's adoption, realizing a certain request of memory using or one in certain scope by single linear list. This scheme will have some influences on the adjacent block combinations and block separations. Its flexibility is also inferior in some cases. Another derivative is partner system in particular separating and combining ways. Partner system has achieved extensive applications, but serious internal fragment problems exist and there's some certain restricts on adjacent blocks combination that will possibly induce rank-connecting operations. Technique of delaying combination can upgrade DMM efficiency that uses partner system.

Some DMMs adopt other more complicated data structures, such as ordered binary tree, Descartes tree. Bit-map reflection gains some applications as well.

Dynamic memory management scheme based on CACHE we propose applies for DMM that is first-fit ordered by address. The address-ordered linear list first-fitting DMM can enhance procedural locality and decline memory fragments tremendously, so it has been researched and adopted prevalently. It is researched that DMM may be remarkably enhanced, especial memory-releasing efficiency, by caching recently freed blocks. The following 3 sections of this paper furnish data structure and algorithm of DMM, evaluation result and analysis based on CACHE.

2. DYNAMIC MEMORY MANAGEMENT BASED ON CACHE

It has been mentioned that DMM first-fitted by address-ordered linear list has gained most extensive researches and applications. But problems also exist in that its flexibility is inferior. If stacks DMM manages are badly large and usages of memory are terribly frequent, the linear list will become so long that seeking in linear list is time-consuming when memory blocks are allocated and released each time: when memory releases, seeking linear list is required to locate the position memory blocks insert in order to keep it address-ordered; when memory applies, it is also required to search suitable memory blocks by traveling whole linear list.

In order to improve efficiency of DMM, a CACHE mechanism is introduced which is upgraded from traditional DMM algorithm using linear list first-fit. It buffers latest released memory blocks by CACHE, which declines searching time tremendously when memory is released and promotes efficiency especially the memory releasing one.

Our method is: two free block linear lists——free-list and

cache-list—are maintained in DMM, whose free blocks are address-ordered. When memory blocks are released, it will be released into cache-list by DMM; when memory blocks are using, DMM searches cache-list at first, if searched, it returns the blocks in CACHE, otherwise, all blocks in cache-list will be combined into free-list. And it searches when combining. If it fails, more virtual memory spaces will be required from operation system.

Dynamic memory allocating and releasing algorithms based on CACHE are as follows in details:

Algorithm 1:

Allocating algorithm. This algorithm seeks a memory block by first-fit strategy whose size is equal to ReqSize at least. And it determines whether separated or not by a threshold T. The following procedure has supposed that free-list and cache-list are all ordered by address escalating.

```

Step 1. /* Initiation */
pFound := nil; pCache := cache_list;

Step 2. /* Search cache-list */
while pCache ≠ nil do
  if SIZE(pCache) ≥ ReqSize then begin
    pFound := pCache;
    if SIZE(pFound) - ReqSize > T then
      Split pFound
    return pFound; end;
  else
    pCache := NEXT(pCache);

Step 3. /* cache-list fails to search , combine cache-list and
free-list, search at meantime */
pCache := cache_list; pFree := free_list; cache_list := nil;
while pCache ≠ nil do begin
  if IS_ADJACENT(pCache, pFree) then
    Combine adjacent blocks: pCache and pFree;
  else if ADDRESS(pCache) > ADDRESS(pFree) then
    begin
      if SIZE(pFree) > ReqSize then pFound := pFree;
      pFree := NEXT(pFree);
    end
  else begin
    Insert pCache into free_list before pFree;
    pCache := NEXT(pCache);
    if SIZE(pFree) > ReqSize then pFound := pFree;
  end;
end;
if pFound = nil and pFree ≠ nil then
  Search nodes after pFree by first fit;
if pCache ≠ nil then
  Put pCache and nodes behind after node pFree;
if pFound ≠ nil then begin
  if SIZE(pFound) - ReqSize > T then
    Split pFound;
  return pFound; end;

Step 4. /* Fails to search in cache-list and free-list , then
allocate virtual memory spaces from OS*/
Apply for virtual memory space from OS;
Spitted into two blocks by ReqSize;
Put one into free_list;
return blocks belong to ReqSize;

```

Algorithm 2

Releasing algorithm .Release block pBlock into stacks .If next adjacent block is free when releasing, combine them, else release pBlock into cache-list, keeping escalating order by address.

```

Step 1. /* get adjacent block address */
pNext := NEXT(pBlock);

Step 2. /* */
if pNext ≠ nil and BLOCK_IS_FREE(pNext) then begin
  Combine pBlock with pNext;
  Substitute pNext with combined blocks;
  return;
end;

Step 3. /* If there's no succeeding blocks or succeeding
blocks are not free, release it into cache-list */
pCurrent := cache_list;
while pCurrent ≠ nil and
ADDRESS(pCurrent) < ADDRESS(pBlock) do
  pCurrent := NEXT(pCurrent);
if pCurrent ≠ nil then
  Insert pBlock before pCurrent;
else
  Insert pBlock after cache_list;
return

```

Only combination of succeeding adjacent blocks is considered when we design the algorithms. Virtually, this method of simplified analysis is adopted in many common DMM.

3. EFFICIENCY ANALYSIS AND ASSESSMENT

Theoretically analyzing, it is argued that efficiency will be improved conspicuously when memory block from algorithm 2 is released into cache-list rather than into free-list, because what cache keeps is the latest released free blocks and blocks in cache are all allocated to free-list when failure each time in distributing cache searching. So, the number of nodes m is far less than n of free block, thus, it's faster to insert in specific place in cache-list. Considering the worst, in algorithm 1, it will combine cache and free list in that the two lists are sequential according to address when cache fails to search usable memory block. The incorporation entails the search of

n nodes plus the expenditure $\frac{m^2}{4}$ that keep address-ordered

when m blocks release to cache, which still superior to the

reordered expenditure $\frac{m^2 + (2n-1)m}{4}$ that directly release

to free-list. After introduction of cache, conventional operation of adjacent blocks' incorporation doesn't subject to any influence and will not increase debris of memory; meanwhile, this scheme inclines to give priority to repeated use of the latest released memory block and thus strengthen program locality to some extent.

Modeled assessment is prevalent in DMM efficiency analysis. Firstly, set up model of memory using, then simulate DMM operation process based on this model, come by running efficiency analytical data in simulation process. Foundation of the model that is used by memory can be obtained through certain model conforming to probability distribution that

stems at random, but the precision of this kind of model has never been carefully studied. We adopt the method that traces running process of practical program to give birth to a memory-using model. Therefore, the practical program for which we adopted is a set of 3-dimensional designing system GEMS developed by TsingHua University with C++ language. Through reloading new/delete, we take down all the dynamic memory applications and releasing operation in GEMS, and

drive DMM with it.

In experiments of the first group, a element map and relevant assemble files of pump design are opened/closed, which generated 151674 memory operations (76172 applying and 75502 releasing). Taking the searching times in list when DMM applies and releases requests as a index of assessment, results are as follows:

	Conventional DMM	DMM based on CACHE
Memory applying (76,172)	22,876,591	44,169,089
Memory releasing (75,502)	233,114,450	173,295,303
Overhead in all	255,991,041	217,464,392

Figure1 experiments of first group

In another group of tests, more operations have been done in GEMS, including operations that creating elements assemble map, engineering map and open/close file. 1659501 memory

operations (862590 applying and 796911 releasing) are generated. The results are as follows:

	Conventional DMM	DMM based on CACHE
Memory applying (862,590)	432,169,456	271,256,661
Memory releasing (796,911)	648,626,016	154,006,504
Overhead in all	1,080,795,472	425,263,165

Figure2 experiments of second group

We have accomplished two experiments, expecting to get various sequences of memory usage, which includes size distribution of applying objects, TTL and distribution of arriving time. In first group of experiments, opening and closing files in sequence makes memory applying and memory releasing concentrated. But in second group of experiments, what are accomplished are GEMS conventional drawing operations. Access applying and releasing operation are crossed; therefore, a shorter cache-list is kept so that efficiency of memory releasing is promoted remarkably. It can be found that the scheme based on CACHE in second experiment upgrades by 76% when releasing and the counterpart in first experiment upgrades by 26%. The efficiency improvement in aspect of memory allocating of DMM based on CACHE will be influenced by the model of memory repeatedly using. Considering the case that the size memory applies for each time is liable to be larger than that released after every preceding memory request, most of it can be realized through the process "Search cache-list—Fail—Combine cache-list and free-list—Search free-list" so that memory allocating expenditure expands. For example, results of experiments in first group show that DMM memory allocating efficiency has decreased by 93% after the addition of CACHE and that in second group, 37%. From the view of hit-rate when memory is allocated, it is 27% (20521) in first group of experiments, which is dramatically lower than that in second group of experiments, 36% (311503). All performances concerned, DMM based on CACHE has been improved by 15% and 60% respectively in two groups of experiments.

4. CNOCLUSION

The high efficient DMM is becoming more and more attractive. Because the using of object-oriented program language makes the dynamic storage and management become more and more important in software systems. During the past 40 years of research and application, researchers have raised many algorithms of DMM in linear list, partner-system, tree, position mapping and so on. Especially, the DMM of orderly matching line-table according address could enhance the part of programming and decrease fragments of CPU. It is used and researched widely. This paper describes a method of dynamic storage and management that uses CACHE. It is suitable for DMM that maps according address and buffers the late released fragment of CPU to improve DMM and the releasing efficiency of CPU. The method is proved to be efficient by theory and practical use. The future work includes the research on the timing ability of DMM based on CACHE and the realizing of automatic collecting of memory garbage.

5. REFERENCES

- [1]P. R. Wilson et al., Dynamic Storage Allocation: A Survey and Critical Review. Proc. of Intl. Workshop on Memory Management, Sep., 1995, Vol(986).
- [2]Woo Hyong Lee et al., Evaluation of a High-performance Object Reuse Dynamic Memory Allocation Policy for C++ Programs. Proc. of The Fourth Intl. Conf. on High Performance Computing in the Asia-Pacific Region, May, 2000, Vol(1):14-17.
- [3]D. E. Knuth. The Art of Computer Programming, chapter Information structures (chapter 2 of volume 1), Boston,

- MA, Addison–Wesley, 1973.
- [4]P. R. Wilson et al., Memory Allocation Policies Reconsidered. Technical Report, University of Texas at Austin, 1995.
 - [5]B. Zorn, D. Grunwald. Evaluating Models of Memory Allocation. Technical Report CU-CS-603-92, University of Colorado, Jul., 1992.
 - [6]P. R. Wilson, Uniprocessor Garbage Collection Techniques. In Proc. of Intl. Workshop on Memory Management, St. Malo, France, September 1992, p.p. 1-42.
 - [7]Doug Lea. A Memory Allocator. <http://g.oswego.edu/dl/html/malloc.html>.
 - [8]I. Puaut. Real-Time Performance of Dynamic Memory Allocation Algorithms. Proc. of 14th Euromicro Conf. on Real-Time Systems, 2002, p.p. 41-49.

The Performative Design of Distributed Storage Agents

Zhu Yong^{1,2}, Zhang Jiangling²

¹Wuhan Institute of Science and Technology, Wuhan, 430073, China

²Huazhong University of Science & Technology, Wuhan, 430074, China

Email: zh8871@public.wh.hb.cn zy@wuse.edu.cn Tel: 027-62318126

ABSTRACT

The distributed storage aims at the high performance and high availability. Data block is guaranteed to access in high speed and dependably in bottom layer of volume management, and data structure of efficiency and security is implemented in the middle layer of file system. The resource is redistributed by dynamic process control and load data collection through distributed storage agents.

The model of distributed storage management based on agents with autonomy and social ability in cooperative work for heterogeneous sources is presented in the paper. And the objective function is setup to optimize the resource scheduler. Finally, the implementation of agents and their communication performative in distributed environment is put forward.

Keywords: Distributed Storage, Agent, Performative

1. INTRODUCTION

Enterprises and units manage and storage the massive information by any kind of means in the information era. But they have found that they are between two fires because of knowledge blast. One side, the ability of data management do not equal to its ambition according to the model of traditional service-centered both in volume and speed of storage. On the other hand, network administrators think it is heavy-laden to integrate the heterogeneous sources. The technology of distributed storage and agents emerge as the times require along with the developing of network. The industry leaders issue their solutions of information storage continuously, such as the AutoIS by EMC, VCS, VVM and VFS by Veritas. The architecture of NAS and SAN based on the technology of distributed storage and agents is accepted by users to replace the DAS, Other stand for iSCSI, FC, VIA and InfiniBand will be applied soon.

The technology of distributed storage that enables largescale flexible resource sharing among dynamic virtual organisations emerges as the times require along with the developing of network. An essential component of distributed storage infrastructure software is the agents layer, which acts as middleware between its resources and its applications. An agent-based resource management system for distributed storage is introduced to address the scalability, availability and adaptability. An agent-based distributed storage system, in view of agents with pro-activeness, autonomy, social ability and mobility dealing with distributed and heterogeneous objects, couples the networks techniques with a scheduling algorithm designed to manage the distributed storage resource. It utilises the agent-based methodology, where each agent acts as a representative for a local storage resource and considers this resource to be its high performance computing capability. Agents cooperate to perform service advertisement and discovery, thus providing the base services with which to

manage and schedule applications over available distributed storage resources. The performance of these agents can be improved by using a number of different optimisation strategies.

A number of recent agent-based distributed projects have utilised existing distributed computing technologies such as CORBA and Jini. While CORBA and Jini are well suited to their original design goals, they are designed for developing high performance computing applications. Specialised agents contain behavioural rules that can be modified based on their interaction with other agents and the environment in which they operate. The Distributed storage agents use a hierarchy of agents for both service advertisement and discovery, and integrate these with a performance prediction based scheduler. The elementary goal is to schedule the tasks and to manage the resources of distributed storage system to transfer data efficiently and reliably, otherwise to integrate and to manage the heterogeneous sources. Finally the architecture of agents and the implementation of their communication performative of distributed storage are presented in this paper.

The paper is organised as follows: the "The Distributed Storage Agents" is presented in Section 2; Section 3 essentializes the "The System of KQML and CORBA"; Section 4 describes the "Performative Implementation of Distributed Storage Agents" and the paper concludes in Section 5.

2. THE DISTRIBUTED STORAGE AGENTS

2.1 The architecture of storage

The system of distributed storage consists of the physical construction and the software for them. Now the architecture of network storage is acknowledged, such as DAS (Direct Attached Storage)、NAS (Network Attached Storage) and SAN (Storage Area Network)。DAS is just a storage device attached to the extended interfaces of service and client. And NAS is a kind of special storage service building in operating system and providing services of file share across platforms. SAN focuses on data with the scalable network topology and high-speed fiber channel to switch nodes among SAN. Data manage is concentrated in an absolute storage net to share data, to optimize management and to expand scale seamlessly.

2.2 The definitions of distributed storage agents

The distributed storage system should not be sensitive to its topology and focus on the communication of information and transfer of data through the distributed storage agents that are defined as follows:

SDA (Storage Device Agent) : The primary tasks of SDAs which are in the bottom of the architecture are that the access request for RAID is implemented and the result, as well as the performance statistic is also return to QBAs. SDAs must implement the coherence among heterogeneous sources and provide the storage services to make the data consistency in

block level of various storage devices for upper agents. IOA (I/O Agent) : They take charge the transfers of data and control information through nets and channels. The great character of agents is that they cooperate with each other to fulfill tasks, such as remote I/O, parallel I/O and abstract I/O. They implement normal I/O certainly.

QBA (Queue Buffer Agent): Their role is as storage buffer in the middle of the architecture. QBAs provide the foundation of task schedule for the upper agents RPAs by receiving the messages of SDAs and IOAs. Otherwise they can callback tasks, reorder the queue and so on according to the character of queue.

FSA (File System Agent): They use a LDAP (lightweight directory access protocol) as well as DNS to implement a three-tiered naming system. i.e. Context objects map context names to OIDs(object identifiers), which are location-independent identifiers that include an RSA public key. Then, A OID is mapped to OA (object address) for communication purposes. An OA consists of an IP address and port number.

RPA (Resource Management & Process Schedule Agent): They are kernel agents and consist some components, such as a resource state information database and scheduler which maps requests to resources. RPAs parse and process the RSL (resource specification language) that outline job requests. They also enable remote monitoring and managing of jobs already created and update directories with information regarding the availability of the resources they manage. RPAs provide the local component for resource management and are responsible for the set of resources operating under the same site-specific allocation policy. Such a policy must be adaptive and will often be implemented by objective function to improve the system performance.

STA (Security & Fault Tolerance Agent): STAs employ an authentication system known as the GSI (generic security service) using the RSA encryption algorithm with public and private keys and rely on an X509 certificate provided by the user in their directory that identifies them to the system. The main detection of a fault service by STAs is the Heartbeat Monitor that enables a process to be monitored and periodic heartbeats to be sent to one or more monitors.

UIA (User Interactive Agent): They interact with user directly implemented in WEB and JAVA. The resources management and task scheduler are intervened by user command via UIA considering the complexity and scalability, though UIAs can work in autonomy without intervene in most cases. Otherwise UIAs provide services of graphic output for network storage informations.

3 . THE SYSTEM OF KQML AND CORBA

KQML (Knowledge Query and Manipulation Language) is both a message format and a message-handling protocol to support run-time knowledge sharing among agents. KQML focuses on an extensible set of performatives, which defines the permissible "speech acts" agents may use and comprise a substrate on which to develop higher-level models of inter-agent interaction such as contract nets and negotiation. In addition, KQML provides a basic architecture for knowledge sharing through a special class of agent called communication

facilitators that coordinate the interactions of other agents. The ideas that underlie the evolving design of KQML are currently being explored through experimental prototype systems that are being used to support several test beds in such areas as concurrent engineering, intelligent design and intelligent planning and scheduling.

As a communication language for intelligent information agents, KQML draws on work in both distributed systems and distributed AI and offer a level of abstraction that should be useful to both. Current KQML implementations have used standard communication and messaging protocols as a transport layer, including TCP/IP, email, Linda, HTTP, and CORBA. As standards in this area evolve and new standards are introduced. The contribution that KQML makes to Distributed AI research is to offer a standard language and protocol that intelligent agents can use to communicate among themselves as well as with other information servers and clients. The independence of the communication and content languages affords a flexibility that is quite useful. In designing KQML, the goal is to build in the primitives necessary to support all of the interesting agent architectures currently in use.

A typical of KQML Software Architecture [9] is illustrated in Fig. 1.

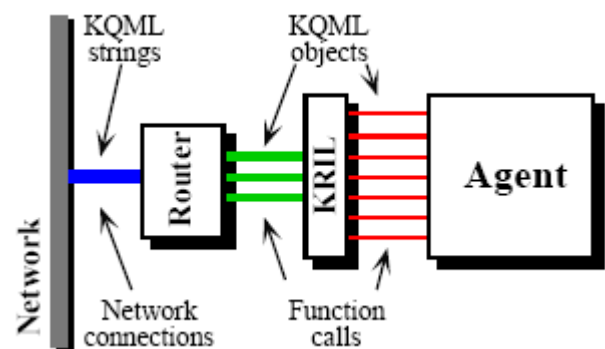


Fig. 1 KQML Software Architecture

KQML will be more important in building the distributed agent-oriented information systems of the future. The design of KQML has continued to evolve as the ideas are explored and feedback is received from the prototypes and the attempts to use them in real test bed situations. Furthermore, new standards for sharing persistent object-oriented structures are being developed and promulgated, such as OMG's CORBA specification and Microsoft's OLE 2.0. Should any of these become widely used, it will be worthwhile to evolve KQML so that its key ideas the collection of reserved performatives, the support for a variety of information exchange protocols, the need for an information based directory service can enhance these new information exchange languages.

CORBA (Common Object Request Broker Architecture) is solidly grounded in fundamental object-oriented programming and is based on a client-sever model of distributed computing. The client of a CORBA object acquires its object reference and uses it as a handle to make method call, as if the object is located in client's own address space. The Object Request Broker (ORB) is responsible for all the mechanisms required to find the object's implementation, prepare it to receive the request, communicate the request to it, and carry the replay (if any) back to the client. The core of the CORBA architecture is

the ORB that acts as the object bus over which objects transparently interact with other objects located locally or remotely. Not only does the broker provide common services, including basic messaging and communication between client and service, It also insulates the application from the specifics of the system configuration, such as hardware platforms and operating systems, network protocols, and implementation language. To invoke operations on a remote distributed object, a client must know the interface offered by the object. The interface, composed of operations and the types of data that required to be passed to and from those operations, is defined in the OMG Interface Definition Language (IDL).

CORBA is suited such challenge in several aspects: Different system platforms and programming languages; Coexistence of client-server or mainframe oriented system application; Lack of a well-defined architecture; Conflicting data formats and semantic definitions.

4 . PERFORMATIVE IMPLEMENTATION OF DISTRIBUTED STORAGE AGENTS

4.1 Definition of agents

Agents are described by IDL in CORBA as follows:

```
<Agent>::= ( <BASIC>, <STATUS>, <COMMUNICATION>, <SERVICE> )
```

Where <BASIC> includes class name, object name, created date and attribute; <STATUS> includes current status, information ability, service ability and energy; <COMMUNICATION> includes message request, processor and message return communicating by KQML. There are different <SERVICE> for different agents described as follows:

```
<SERVICE_SDA>::= (<READ>, <WRITE>, <STRIP>)
<SERVICE_IOA>::= (<INIT>, <SEND>, <RECEIVE>)
<SERVICE_QBA>::= (<QUEUE>, <BUFFER>, <DESCRIPTO
R> )
<SERVICE_FSA>::= (<OPEN>, <CLOSE>, <READ>, <WRITE
> )
<SERVICE_RPA>::= (<CHANNEL>, <RAID>, <JOB>, <TASK
> )
<SERVICE_STA>::= (<GSI>, <RSA>, <X509>)
<SERVICE_UIA>::= (<BACKUP>, <RECOVERY>, <VIEW> )
```

Applications based on agents employ KQML performative to express communication actives. KQML not only passes some kind of language content, but also message property. It supports the middle-ware integration also.

4.2 Communication performative

The example of communication performative among RPAs and UIAs is as follows:

```
( ask-one
  : sender UIA
  : content "RSL Based Command"
  : receiver RPA
  : reply-with Par1 , Par2...
  : language C )
( answer-one
  : sender RPA
  : content "Scheduler"
```

```
: receiver UIA
: in-reply-to Ret1 , Ret2...
: language C )
```

"ask-one" and "answer-one" are the performative pair of request and response among RPAs and UIAs. UIAs request RPAs by RSL with Par1 , Par2... parameters and RPAs response UIAs by scheduler with results of Ret1 , Ret2... parameters. RSL and scheduler defined in detail as follows:

```
<RSL> ::= ( <BACKUP>, <RECOVERY>, <CHANNEL>, <RESOURCES> )
```

```
<Scheduler> ::= ( <JOB>, <TASK>, <BUFFER>, <BLOCK>, <FILE>, <LOAD> )
```

The above definitions can be implemented by storage semantics:

```
<CHANNEL> ::= ( <NIC>, <SCSI>, <FC>, <IDE> )
<RESOURCES> ::= ( <RAID>, <TAPE>, <NAS> )
```

4.3 Communication process among agents

The primary implementations of distribution are CORBA by OMG, COM/DCOM by Microsoft and JAVA RMI by SUN, and besides, PVM and MPI for cluster are also technology.

There are some steps from service request by agents to result return: Agent analyses information records from other agents to demand information; The processor in <COMMUNICATION> transforms the information demand to KQML performative, and IDL stubs of client code the information and produce IDL style request; Generally RPA forwards the request to service agents; IDL skeleton decodes the IDL style request and produce KQML style description which is transacted by service agents; Then, IDL skeleton codes the result and exception and produce IDL style results; The results are transmitted by service agent to client agent; Finally, IDL stubs of client decode the result to KQML style description.

4.4 Selection process among agents

The most optimal agent among agents is selected according to the rules and knowledges in definition of agents, called objective function (i.e., information ability, service ability and energy). The rule is described as follows:

```
do
  if Match( pAgent-> Status . Service , Service )
    if( pAgent-> Status . Ability > Ability )
      {
        Ability = pAgent-> Status . Ability ;
        ptrAgent = pAgent ;
      }
  pAgent++ ;
  while pAgent != NULL ;
  task(ptrAgent) ;
```

4.5 The objective function

Objective function can be expressed as:

$$E = f(x) + \sum_{\alpha=1}^P C_{\alpha} \rho[g_{\alpha}(x)]$$

Where P is the number of constraint, C_{α} and $\rho[g_{\alpha}(x)]$ are the constant and function for constraint respectively.

$$y = f\left(\sum_{i=1}^n w_i x_i - \theta\right)$$

is the typical model of neural.

5 . CONCLUSION

The architecture of network storage agents is of distribution and hierarchy. In the view of architecture, there are storage devices, channels, NICs and hosts which are managed by corresponding agents from bottom to top layer; From point of view of storage medium, there are CACHE, RAM, disk array and tapes which are administered by relevant policy from inner to outside. It is fundamental for the agent-based system to provide such services as volume management, share storage management and virtual devices management. The services of online data, disk fault recovery and security are also offered.

The blue print of distributed storage based on agents is put forward in this paper by the performative design of distributed storage agents implemented in KQML and CORBA to improve the scalability, availability and adaptability.

There is not an almighty system now because of complexity and standard of distributed storage system. The solution in the paper is improved by learning in operating, and there are much more works to do.

6 . REFERENCES

- [1] Rajkumar Buyya . High Performance Cluster Computing Architectures and System . Prentice-Hall, Inc, 1999
- [2] Junwei Cao, Stephen A. Jarvis, Subhash Saini, Darren J. Kerbyson and Graham R. Nudd . ARMS: An agent-based resource management system for grid computing . Scientific Programming 10 (2002) 135 – 148
<http://www.dcs.warwick.ac.uk/~saj/papers/arms.pdf>
- [3] Xia Manmin , Li Huaicheng . Intelligent Software Agent for Application in Distributed Information Management. Journal of Nanjing University of Aeronautics & Astronautics , Vol32, No5, Oct. 2000
- [4] LIU Bin , WANG Lan-shao , WANG Hao-jun . COMMUNICATION ARCHITECTURE OF DISTRIBUTED AGENTS BASED ON CORBA . MINI —MICRO SYSTEM , Vol22 No6 , June 2001
- [5] Genesereth M R , Ketchpel S P . Software agent . Communication of the ACM , 1994 , 37 (7)
- [6] IBM Corp. Tivoli Storage Manager Managed System for SAN Storage Agent User's Guide. July 2000. docs.rhrz.uni-bonn.de/tsm_pdf/msyssan/anrcst40.pdf
- [7] VERITAS Software Corporation. VERITAS Cluster Server™ Storage Agent for NetworkAppliance. 2001. http://eval.veritas.com/webfiles/datasheets/netapps/vcs_agents_netapps_datasheet.pdf
- [8] Yannis Labrou, Tim Finin. A Proposal for a new KQML Specification. February 3, 1997.
<http://www.cs.umbc.edu/kqml/papers/kqml97.pdf>
- [9] Tim Finin, Richard Fritzson. KQML as an Agent Communication Language.
<http://www.cs.umbc.edu/kqml/papers/kqmlacl.pdf>
- [10] CORBA Scripting Language Specification Version 1.1. February 2003.
<http://www.info.fundp.ac.be/~ven/CIS/OMG/facility/corba%20scripting%20language.pdf>
- [11] Yanfeng Gong. CORBA Application in Real-Time Distributed Embedded Systems. Survey Report , ECE 8990 Real-Time Systems Design, Spring 2003.
<http://www.ece.msstate.edu/~jwbruce/rtes/corba.pdf>
- [12] Thomas R. Obritz, Florian Schintke, and Thorsten Schutt . Architecture of the Resource Management System of WP4 (Fabric Management) , September 24, 2001.
<http://hep-proj-grid-fabric.web.cern.ch/hep-proj-grid-fabric/architecture/rms.pdf>
- [13] Jin Xiong, Ninghui Sun. The Scalability and Availability in Cluster Resource Management, HPC-Asia'2001 .
<http://www.ict.ac.cn/xueshu/2001/h028.doc>
- [14] O. F. Rana, D. W. Walker, and Y. Huang . Architecture of an Intelligent Resource Management System . December 15-16, 1999.
<http://www.cs.cf.ac.uk/User/David.W.Walker/papers/ukplan-final.ps>
- [15] Ursula Maier, Georg Stellner. Distributed Resource Management for Parallel Applications in Networks of Workstations, 1997.
<http://citeseer.nj.nec.com/cache/papers/cs/584/http:zSzzSzwwwbode.informatik.tu-muenchen.de:zSarchivzSzartikelzSzhpnc97zSzhpnc97.pdf/maier97distributed.pdf>
- [16] William Michael Jones. Scheduling and Resource Management in Computational Mini-Grids, July 1, 2002.
<http://parlweb.parl.clemson.edu/~wjones/research/draft.pdf>
- [17] Massimo Sgaravatto. First ideas for a Resource Management Architecture for Productions.
<http://server11.infn.it/workload-grid/docs/20000628-sgaravatto-rm.pdf>
- [18] Design, Implementation and Evaluation of Resource Management System for Internet Servers, 2002.
<http://www.anarg.jp/achievements/web2002/pdf/proceedings/tak-okmt02SIGCOMM-ResourceManagementSystem.pdf>
- [19] James B. White, Richard A. Alexander. Lessons Learned from Proprietary HPC Cluster Software, June 27, 2001.
http://www2.ccs.ornl.gov/staff/trey/pubs/lessons_ppt.pdf
- [20] Sun Microsystems, Inc. A New Open Resource Management Architecture in the Sun HPC ClusterTools™ Environment, November 2002.
<http://www.sun.com/solutions/blueprints/1102/817-0861.pdf>
- [21] Marco AurÉlio Stelmar Netto_ , CÉsar A. F. De Rose_. CRONO: A Configurable Management System for Linux Clusters, October 23-25, 2002.
http://www.democritos.it/activities/IT-MC/cluster_revolution_2002/PDF/21-DeRose_C.pdf
- [22] Zhang Liming . Models and Applications of Artificial Neural Networks . Fudan University Press , 1993 , 7
- [23] Dan Kidger. The Quadrics Products : a view from an HPC user, Machine Evaluation Workshop 2001.
http://esc.dl.ac.uk/Cdrom-s/12th_Machine_Evaluation_Workshop/Html/Talks/Kidger_Quadrics.pdf
- [24] Duncan Roweth. Quadrics Interconnect, SC2001 Denver.
<http://www.c3.lanl.gov/~fabrizio/talks/quadrics.pdf>

A New Solution Scheme of NAT and IPSec Protocol Compatibility Problem Based on IP Tunnel*

Li Fangmin^{1,2}, Xue Ligong², Wang Runyun¹, Zhang Huifu^{1,2}

¹School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan, 411201

²School of Information engineering, Wuhan university of technology, Wuhan, 430070

Email: lfm68@sina.com Tel: 008613973255023

ABSTRACT

IP_in_IP tunnel technology is used to expand NAT functions to solve the compatibility problem between IPSEC and NAT protocol at present. Relative to the solutions existing at present, this scheme only needs to expand a NAT tunnel function in the present NAT function module, and does not need to change existing NAT and IPSEC infrastructure, which can solve the compatible problem very good. In a word, the scheme has very good application prospects.

Keywords: NAT , IPsec , IP in IP , Tunnel NAT

1. INTRODUCTION

NAT [1] (Network Address Transfer) and IPsec (Internet Security Protocol) [2] are two protocols which are widely applied in IP network. NAT works by modifying address and the ports of upper protocol, but because of the function of the integrality and privacy of the data which is provided by IPsec, the data is thought as illegal and refused to be received by the receiver once the modifying is executed. In order to solve the problem, there are two main solutions at present: RSIP[3~4] and UDP encapsulation [5~7]. The first solution is that substitutes NAT with RSIP and offers support to IPsec by additional RSIP. The second one is that support the existing NAT system by modifying IPsec protocol. Since the first solution demands modify the terminal system, it is difficult to realize in a short time. The second solution demands modify existing IPsec system and the thought is too complex.

This paper introduces the thought of IP_in_IP tunnel[8] by the research of NAT and IPsec protocol compatibility problem, designs a kind of new solution, which only demands add a kind of NAT function in existing NAT gateway—tunnel NAT(Say strictly, the thought doesn't belong to NAT category, but it solves the function which is similar with NAT, so it also can be classified with NAT). The solution can solve the problem of compatibility problem of IPsec and NAT without modifying the existing NAT and IPsec system..

2. PROBLEM DESCRIPTIONS

Let us take a network topology with two net points for example, we look respectively the typical application of NAT and IPSEC, and then elicit the question by combining them.

2.1 Typical Application of NAT

Fig.1 shows NAT scheme which is adopted by most only own

private IP sub-network, private address is distributed designedly to affiliated net point by the network management organization who own those net points (The advantage of doing that is the address needn't change if each net point need interconnection through hired wire). Each sub-network only need ISP connected and provides a common network address, and it can make net point 1 and point 2 visiting Internet through the NAT function of R1 and R3. Private sub-networks exchange their data through the server of Internet. What is noticeable is that two special departments separate with private network through automatically using Router R2 and R4 because of the problem of data sensitivity (Make some visiting lists in the Router in order to prevent the other host computers outside the special department visiting the host computer in the network).

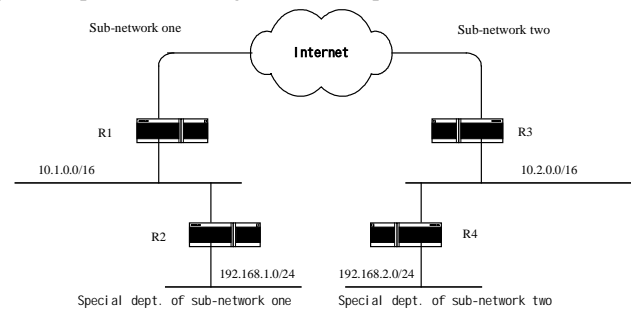


Fig. 1 A Representative Application Case

2.2 The Elicit of Problem

Two special departments decide to interconnect their network with VPN gateway because of the following reasons: The special department data exchange frequently, so it need exchange visit opponent host computers directly; Though the scheme of Fig.1 make it reduced that the probability of directly visit the host computer of special department from net point and INTERNET, the data sent out from the special department are still proclaimed in writing. It is still possible for those people who have ulterior motives to wiretapping the secret data.

It don't change network topological in this way, and replaces R2, R4 with VPN (Virtual Private Network) gateway based on IPSEC. But the problem appears. Because of dynamic address transition will take place in R1 and R3, which makes source address and upper protocol terminal changed, and results in the data, which are provided by IPSEC protocol of R2 and R4, cannot pass, and the network of two special departments cannot interconnect.

3. PRINCIPLE OF TUNNEL NAT

* This work is supported by China Hunan Education Science Foundation #02A049 and Hunan NSF #03JJY1012.

A. IP in IP Encapsulation

IP in IP encapsulation, this is to say, inserts IP header outside of the data package of IP, which is showed in Fig.2. The setting of new IP head is as following:

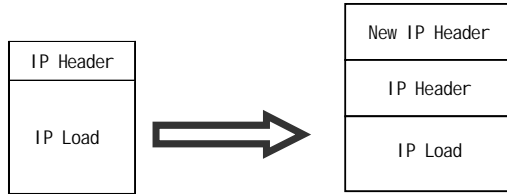


Fig.2 IP in IP Encapsulation

Edition: 4

ToS: copy from the original IP header

IHL: the length of the new IP header with 32 bit as unit

Length of data: sum of length of original data and new IP header

Identifier, symbol and fragmentation offset: The setting of those regions is same as the produce of IP protocol, what is worth paying attention to is that DF bit (don't fragment) should copy from the response of original IP. If the original IP don't allow to fragment, then the new IP is also don't allow.

TTL: TTL of original IP subtract 1

Protocol number: 4

Checksum sum: calculate according to the new IP head

Source address and destination address: source address is the address of host computer which inflictsions IP in IP, and destination address is address of the host computer which ultimately unbind IP in IP encapsulation.

B. Principle of tunnel NAT

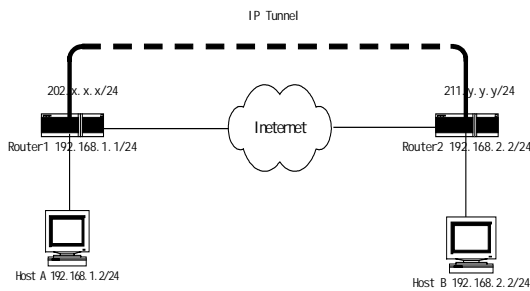


Fig. 3 IP Tunnel

The thought of tunnel NAT is that utilizes the protocol of IP in IP tunnel, builds a IP in IP tunnel between gateway and gateway, which makes IP network in both terminal of gateway exchange data pellucidly through tunnel. The NAT function can parallel exist with existing NAT, which NAT function an entering data package will be adopt, can be defined by using strategy rules. In Fig.3, host computer A and B only have private address, while ROUTER 1 and ROUTER 2 have common net address and the function of tunnel NAT. We will take a data stream from host computer to host computer in Fig.3 for example in order to explain the working course of tunnel NAT.

Host computer A builds the data package to host computer B, the destination address of IP head is 192.168.1.1, and the source address is 192.168.1.1;

The data package is transmitted to gateway Router1;

NAT function module of Router1 exerts tunnel NAT to the data package according to NAT strategy— - inserts new IP head, the destination address of the new IP head is 211.Y.Y.Y. and its source address are 202.X.X.X.;

The new data package arrives at Router2 through Internet Router; NAT function module of Router2 deletes the IP head of the new data package according to the strategy; Router2 transmits the original data package to host computer B according to the IP head of the original data.

The return course of data package from host computer B to host computer A is same as it. Fig.4 shows the course of data stream from host computer A to host computer B.

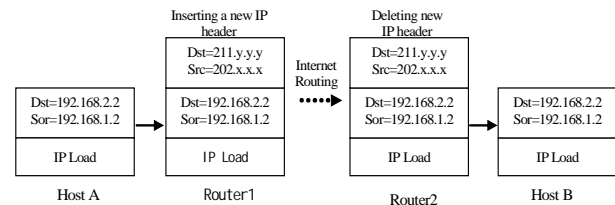


Fig. 4 Example of Tunnel of NAT

Tunnel NAT doesn't change the content of the original data package, and needn't maintain the address relation-mapping table as the traditional NAT, and it also can use by nesting. The disadvantage is that each tunnel need add corresponding strategy, but considering that the tunnel is usually used in the connection between network and network, and the strategy is relatively regular, so it still can be acceptable.

4. THE SOLUTION SCHEME

Now let us come back to the problem which was put forward in 2.2, at present, the contradiction of NAT and IPSEC protocol compatibility is that NAT need to revise the data of original data package, while IPSEC protocol system doesn't or partly allow revise the data of original data package. The problem can be solved well if tunnel function be added in current NAT gateway. R2、R4 changes VPN gateway in Fig.1, the data from special department host computer of net point 1 to special department of net point 2 become IPSEC data package through the disposal of R2, the new IP head will be inserted in data package when it through R1 because it was exerted tunnel NAT strategy, transmits the new data package to R3 through IP tunnel, R3 deletes the new IP head of data package according to the tunnel NAT strategy, then the original data package is transmit to R4, the content of original data package didn't any change in the whole course, so it can pass the integrity verification of data package by R4, and R4 will dispose the data package according to the IPSEC system Rule. To IKE, which generally is used with IPSEC at the same time, the above-mentioned method is also suitable.

The following is pseudo - code description of R1、R2、R3、R4 during the course of a data package is transmitting:

R1

```

Ip_input (char * ip_packet) {           /*IP  package  enters
disposal module */
    .....
    If (Ipsec_policy (ip_packet))        /*judges   whether   the
package accords with IPSec exerting strategy according to the
information of source address and so on */
        new_ip_packet = Ipsec_apply (ip_packet);    /*carry
through IPSec disposal*/
    Ip_output (new_ip_packet);           /*transfer disposal program
outside the IP package */
}

```

R2

```

Ip_input (char * ip_packet) {           /*IP package enters the
disposal module */
    .....
    if (Tunnel_nat(sor_ip))              /*judges whether the package
accords with tunnel NAT exerting strategy according to the
information of source address*/
        new_ip_packet = Tunnel_nat_apply (ip_packet); /*carry
through tunnel NAT disposal*/
        Ip_output (new_ip_packet);       /*transfers   disposal
program outside the IP package*/
}

```

R3

```

Ip_input (char * ip_packet) {           /* IP package enters
disposal module */
    .....
    if ( Ip_in_ip(ip_packet) && tunnel_nat(sor_ip)) /*judges
whether the package is IP_in_IP data package and it accords
with tunnel NAT strategy or not according to the information of
IP package */
        new_ip_packet = Tunnel_nat_desencap (ip_packet);
        /*unpacks tunnel NAT */
        Ip_output (new_ip_packet);       /*      transfers
disposal program outside the IP package */
}

```

R4

```

Ip_input (char * ip_packet) {           /*IP package enters
disposal module */
    .....
    if(Ipsec_policy(sor_ip))             /*judges whether the package
is IPSec package or not according to the information of source
address */
        new_ip_packet = Ipsec_desencap (ip_packet);
        /*unpacks IPSec */
        Ip_output (new_ip_packet);       /*      transfers
disposal program outside the IP package */
}

```

5. CONCLUSIONS

This paper contraposes existing compatibility problem between NAT and IPSEC protocol puts forward a kind of solution of tunnel NAT, the solution needn't change anything of the current NAT and IPSEC system, and it can solve well the compatibility problem between NAT and IPSEC protocol.

6. REFERENCES

- [1] P. Srisuresh, M. Holdrege. IP Network Address Translator (NAT) Terminology and Considerations. RFC 2663, 1999
- [2] S. Kent, R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, 1998
- [3] M. Borella, et al. Realm Specific IP: Framework. RFC 3102, 2001
- [4] M. Borella, D. Grabelsky, et al. Realm Specific IP: Protocol Specification. RFC 3103, 2001
- [5] HUTTUNEN A, SIREWALD J, DIXON W. IPSec ESP encapsulation in UDP for NAT traversal [EB/OL]. <http://www.ietf.org/internet-drafts/draft-huttunen-ipsec-sep-in-udp-01.txt>, 2001-03.
- [6] KIVINEN T, STENBERG M. Negotiation of NAT-traversal in the IKE [EB/OL]. <http://www.ietf.org/internet-drafts/draft-ietf-ipsec-nat-ike-01.txt>, 2001-04.
- [7] HUTTUNEN A, DIXON W, SWANDER B. UDP encapsulation of IPSec packets [EB/OL]. <http://www.ietf.org/internet-drafts/draft-ietf-ipsec-udp-encaps-01.txt>, 2001-10.
- [8] W. Simpson. IP in IP Tunneling. RFC 1853, 1995

Some Strategies in the Distributed and Mobile GPS System

Min Peng¹, Yanxiang He², Wensheng Hu³

¹. College of Computer, Wuhan University, Wuhan, 430079, China, hhdawn@public.wh.hb.cn

². College of Computer, Wuhan University, Wuhan, 430079, China, yxhe@whu.edu.cn

³. College of Computer, Wuhan University, Wuhan, 430079, China, xwindx@163.com

ABSTRACT

A distributed and mobile GPS system framework is designed in this paper. Some strategies are utilized in the system including multi-agents mechanism, CORBA middleware, and probe mobile agent and GML geo-spatial data description. A much wider range of mobile devices using wireless link technology are supported with GIS/GPS applications in our system. Moreover, these strategies can implement distributed GIS and be in favor of overcome bottle-neck of CPU and bandwidth of mobile end devices, offering system flexibility and extensibility both on the server side and client side. The implementation of the strategies is described in detail.

Keywords: Multi-Agent, CORBA, GPS, Distributed GIS Database, Wireless Device

1. INTRODUCTION

Mobile devices in the form of portable telephones, pagers, PDA and notebook computers are now commonplace. All these devices can be distinguished from each other by functionality, physics characteristics and destination. But they are currently poorly integrated in GPS application. It is very difficult to categorize mobile devices because their features vary a lot. User with different characteristics may be interested in different geographical information presented on a page and may use different navigational style and different devices such as PDA, Mobile Phone and PC browser.

GPS-GIS integrated systems provide the operators with location, speed, and distance traveled in a certain time and time taken to complete trips, which can be used for automatic billing to make payments in case of hired private vehicles and for assessment of performance of the fleet to ensure public comfort. The GPS-GIS system also monitors idling vehicles and precise destination locations for devising shortest paths for total coverage. The data that is provided by these systems is fed to the logistics and optimization software that various vehicle operators use to manage their operations.

The higher-bandwidth mobile networks like GPRS (General Packet Radio Service) together with Pocket PC Phones, Smart Phones and PDAs will allow more acceptable services. It is possible to develop a range of new and exciting location-based applications. Combined with new technologies like intelligent software agents and user modeling like profiling of personal interests the potential for intelligent location-based services is even higher.

But some questions must be considered during the usage:

- 1) Diversity of terminal devices.
- 2) Distributed and multi-formatted geo-spatial data.
- 3) The wireless bandwidth is much lower than wire network.
- 4) The capabilities such as CPU and memory of mobile devices are quite limited.

- 5) The amount of geographic information is very tremendous.

Aiming at those questions in distributed and mobile GPS system, we provide some strategies, including multi-agents technology, CORBA middleware, probe mobile agent and GML geo-spatial data description. In our system, much wider range of client devices are supported with GIS/GPS applications, including traditional wired/desktop contexts and mobile computing devices such as PDA, mobile phone and notebook, which use wireless link technology. It also can distinguish among adaptivity to the specific devices and to user's profile, and be implemented in distributed GIS.

2. SYSTEM FRAMEWORK

Our distributed and mobile GPS system is a system applied for both wireless devices and tradition desktop computer. The use of Multi-Agent and CORBA can implement the distributed GIS and be in favor of overcome bottle-neck of CPU and bandwidth. The framework of system is described in Figure 1.

User terminal

The wide ranges of terminals from cellular mobile phones, PDA, laptop computers to usual desktop computers are considered in our system. All these devices can be distinguished from each other by functionality, physics characteristics and destination.

It is very difficult to categorize mobile devices because their features vary a lot. User with different characteristics may be interested in different geographical information presented on a page and may use different navigational style and different devices such as PDA, Mobile Phones and PC browser.

So more complicated and wider context should be considered for the delivery of information. Network, protocols, multimedia information type and other services of varying client devices are quite different, which need a flexible and steady mechanism to meet the challenge.

Users supported in our system include wireless enders such as pocket PC with Windows CE operation system, mobile phone with K-Java platform, notebook with normal operating systems and conventional wire PC users. There are different characteristic and adaptable functions to those users.

1) Desktop computer

Based on Internet network, some services are supported by desktop computer end users, such as map displaying, road-network analyzing, and information inquiring and so on. It supports both grid map data and vector data.

2) Mobile devices

Based on limited presentation capabilities and operating systems of different mobile devices, different services are

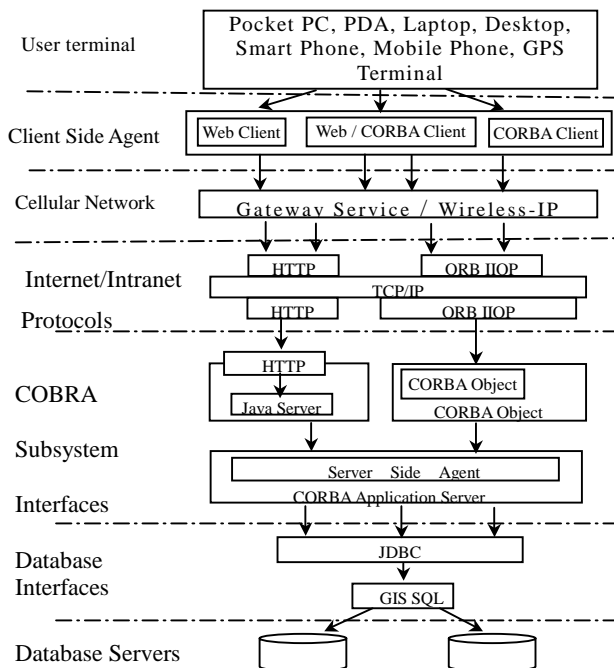


Figure 1. Architecture of Distributed and Mobile GPS System

supplied to them. Most of Pocket PC and mobile phone can interpret the grid map data and pictures such as jpeg, png and gif exactly, but can't deal with and display on vector data directly very well. Generally, mobile services include map displaying, information inquiring and updating, information communication and so on.

Multi-Agents

The architecture relies upon an agent, split into a Server Agent and a Client Agent. Filtering and personalization take place on the server or on the browser, to analyze what is the best.

User Agent serves as middleware between end users and the services offered on the middleware servers. They are software that reside in the client, and provide specific functionality to the end-user. They perform some or all of the following tasks.

- 1) Display content to the end-user. The function is like browser.
- 2) Requesting services from CORBA objects or HTTP servers on behalf of the user.
- 3) Investigate network misbehavior and pinpoint problems and errors.

Server Side Agent is in GIS Application Server, being provided to implement the application service.

- 1) Analyze users' queries and make the optimal strategy.
- 2) Sending mobile agent to sense and detect the status of network traffic between the local server and other remote servers or other clients.
- 3) Fulfill the query tasks sending by client agent.

CORBA subsystem

There are different characteristic and adaptable functions of end users in our system, which need different GIS data and services. At the same time, we expect to communicate information and deal with geo-spatial data between distributed

database and server. In order to provide an implementation language-independent and network protocols-independent framework in which a wide variety of GIS datasets are shared in a distributed, object-oriented, and peer-to-peer fashion, we provide a middleware environment based on CORBA. The detail of the subsystem is described in section 3.

3. KEY STRATEGIES IN SYSTEM

CORBA Utilized in Adapting Distributed Framework

In the system, CORBA subsystem is composed of three function parts: CORBA Object Server, Web Server and GIS Application Server. They fulfill the different function of middleware, and cooperate with Client Side Agents and Server Side Agents that belong to different end user devices and servers.

CORBA Interfaces for GIS Objects: CORBA consists of a core implemented by the various commercially available ORBs (Object Request Brokers) and a number of specified object services and application facilities (CORBA services and CORBA facilities). When an end user invokes an operation, the ORB is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object, and returning any response to the caller.

The Interface Definition Language (CORBA-IDL) and the application programming interfaces (APIs) enable client-server object interaction which in a specific implementation of an ORB. We define a CORBA IDL interfaces based on GISCORBA object attributes and behaviors. The Internet Inter ORB Protocol (IIOP) makes any CORBA ORB instantly usable access the Internet without requiring any additional programming.

There are three interfaces for client, they are feature interfaces, spatial reference system interfaces, and geometry interfaces. On the client side, users access and operate the spatial information such as points, lines and planes through the interfaces.

CORBA Subsystem Framework: To the three parts of CORBA subsystem there are different functions respectively.

To CORBA Object Server, some objects are defined to fulfill the essential functions of CORBA subsystem. Feature Object represents of a real world entity or an abstraction of the real world. It is constructed from geometry objects, attributes (properties), a spatial referencing system, and associated methods. A Geometry object has coordinates that can be mapped to positions in the real world by spatial reference information. A GIS layer is a collection of features and each GIS layer exposes itself to clients through either CORBA Naming or CORBA Trader (Catalog) services (the "White Pages" and the "Yellow Pages"). We can define more CORBA interfaces to provide a rich set of geo-spatial characteristics.

Web Server accepts the requests from Web Client directly, and HTML pages are user interface. The pages are created by Servlets, which obtains services from CORBA subsystem.

GIS Application Server provides services to clients, including Yellow Pages Service, Position Service, Requiring Service, Map Update Service, 3D Mapping Display service and so on. In GIS Application Server, Server Side Agent (SSA) is provided to implement the application service.

The Description of CORBA Implementation:

- 1) Based on CORBA IDL interface, client requests CORBA objects with static stubs compiled in Java, C, or XML;
- 2) The server ORB core is sent to server side application by Client ORB core.
- 3) Object Adapter assigns the requests to Servants of the CORBA objects. Static state invoking method is selected in servant programs, and Static Skeleton is defined by IDL.
- 4) After the execution, Servants return the result to Client Side Agent (CSA).

The abstract model for implementation is described as Figure 2.

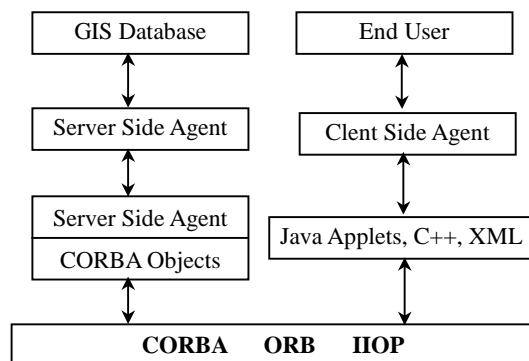


Figure 2. An Abstract Model of CORBA Implement

GML Applied in Geo-spatial Data Description

Spatial information database is an infrastructure of the location-based context-aware service. The database manages the geometric information of the ubiquitous computing environments, such as location of users and surrounding sensor devices.

Although we can use CORBA ORB to deal with the distributed GIS databases described in the previous section, the cost is quite high. If the geo-spatial data have been processed into a common understanding format for interfaces of server and client devices, the cost of services will be decreased greatly and the response of end users' requests will be quicken obviously.

We have developed the kit based on representation of the GIS data based on GML. GML is the standard protocol for encoding spatial data based on XML. The kit can convert various formats GIS data of ArcGIS, Mapinfo into our GML. And we named our kit as Map DB Manager (MDM). The interface of MDM is as Figure 3.

We characterize the MDM with the function bellow:

- 1) Provide an open, vendor-neutral framework for the definition of geo-spatial application schemas and objects;
- 2) Allow profiles that support proper subsets of GML framework descriptive capabilities;
- 3) Support the description of geo-spatial application schemas for specialized domains and information communities;
- 4) Enable the creation and maintenance of linked geographic application schemas and datasets;
- 5) Support the stoppage and transport of application schemas and datasets;
- 6) Increase the ability of organization to share geographic application schemas and the information they describe.

Multi-Agents

Multi-Agents technology is designed to overcome the problem of client devices variety and mobile devices' bottle-neck of CPU and bandwidth, and offer system flexibility and extensibility both on the server side and client side.



Figure 3. An Interface of MDM

Client Side Agent: A much more wider range of client devices are supported as Clients, including traditional wire desktop contexts and mobile computing devices such as PDA, Pocket PC Phone, Smart Phone, and mobile phone, which use wireless link technology. A user can query the service suffered by server side. Before the query sent to server, Client Side Agent (CSA) will fulfill its duties in two stages:

- 1) When a user send a request for some services through Query Interface, CSA does initial parsing and grammar detecting firstly, and distinguishing clients into three types: Web Client, Web/CORBA Client or CORBA Client based on users' query information, and then formulate a list of tasks that will be allocated to server agents.
- 2) Announcing those tasks to server side agents stay in CORBA objects or HTTP Server with additional information about its device capabilities which are some parameters deciding the services that the user can reproduce including sound and images. Those information will decide which server side agent will be selected to help the server to fulfill tasks.

And after the query task is fulfilled at the server side and the services are provided to CSA. CSA displays content to the end-user. Its function likes browser. But before its display, CSA should analyze the device capabilities firstly and then decide its displaying parameters.

Server Side Agent: Once a web server agent is selected from server side, it is staying to fulfill the query task. Its duties are described as following:

- 1) Analyzing users' queries and make the optimal strategy including the response content, size, format and so on.
- 2) Sensing and detecting the status of network traffic between the local server and other remote servers or other clients.
- 3) Based on parameters in section 1 and 2, searching for the appropriate methods for carrying out the task, and cooperating with CORBA objects or HTTP Server to fulfill the task, if the data in database are needed, SSA should start a JDBC.
- 4) Returning the result to ORB.

Client and Server Sides Agents Cooperating:

When a CSA requires a service from others, it selects the most appropriate SSA to cooperate the task. The selection prefers SSA that (1) has the expertise to perform the task, (2) has preciously performed a similar type of task and (3) has been allocated the least number of tasks. Their cooperation is described as following:

- 1) CSA sends a confirm message *accept* (α , β , t) to SSA and the cooperation begins, t is the time. And send a cancel message to other SSA to stop their negotiations.
- 2) Based on the negotiation with CSA, SSA adds some cooperating actions to its initial scheme and executes the task.
- 3) After the complete of task, SSA informs CSA and recalculates its cost.
- 4) Close the cooperation.

Mobile user will query services fulfilled by CSA and server. The query service message will be set up by CSA and send to SSA. Message is decomposed into keywords and location, mainly including the following:

```

QueryDescription
  SessionID(service_name*)
  Title (string*)
  Creator ()
    User_name(string*)
    User_ID()
  Value or ServiceID(string*)

```

The sessionID is a unique identifier for the query of services. The Title is about the topic of the query and the Creator is user name. The value is the actual query string or defined service tag.

We refine the processing from user's service query into task by CSA send to SSA, CSA&SSA representations, computational and processing strategies and collaboration credibility measurements are designed in detail, the following is the partial fro example:

```

AgentProcessing
  Agent()
    Classical_Planner()
  Expression()
    Keyword_list(field*, header*)
    Sring(string*)
    Symbol(symbol_name*)
    Keyword ()
    Keyword_Pair(field_keyword*, field_value*)
    Literal()

```

Mobile-Agent to Detect Traffic Status

During a GPS/GIS application, plentiful data or images should be carried. It is important to aware bandwidth and status of traffic beforehand. Before the communication of client and server, one back-to-back probe packet-pair is sent from CSA to SSA, and CSA receives the bounced packet pair from SSA to compute the available bandwidth. This back-to-back probe packet-pair is implemented by mobile-Agent (MA). Mobile-Agent is computing entities that act on behalf of a principal (User, group, organization) and can autonomously migrate during the execution from one host to another one to continue their operations there. Here we implement it where a Mobile Agent Environment is provided.

- 1) Mobile Agent Manager
It is an agent which is responsible for managing other MA.
- 2) Mobile Agent Transportation Protocols which are defined to

control the transferring of mobile agents.

3) Mobile Agent Naming

It is necessary to provide mobile agents' naming service, which provides the mechanism of tracing mobile agents.

4) Mobile Agent Communication Languages

It is the bridge to cooperation, events transmission and so on among agents.

5) Mobile Agent Security

It is responsible for detecting intrusions as soon as possible and protecting the security of resources of server.

Some special MA software such as IBM Aglets, needs a high environment capability both in client and server side, is difficult to be implemented in mobile devices because it will take more of mobile devices' narrow resources. Our MA mechanism based on Java Applet, and mobile devices side MA is encoded in client side applet. Thus those devices that can run Java Applet can use MA mechanism.

4. CONCLUSIONS

To achieve high performance in GIS/GPS application both with wireless end devices and desktop, CORBA and Agent based cooperative design is used, which can not only distinguish adaptively to the specific device and to user's profile, but also can be applied in distributed GIS. The architecture offers system flexibility and extensibility both on the server side and client side. A prototype of system on PDA is printed in Figure 4..



Figure 4. A Prototype of Mobile GPS on PDA

Future work will focus on implementing more rich services on CORBA ORBs and applying mobile agent mechanism in our system.

5. REFERENCES

- [1] Mowbray TJ, Ruh WA, 1997, Inside CORBA; distributed object standards and applications, Addison-Wesley
- [2] Open GIS Consortium (OGC), 2000. OpenGIS geography markup Language. <http://www.opengis.org/techno/specs>.
- [3] Fuhua Lin, Larry Korba, 2000. Incorporating Communication Monitoring and Control Facility in Multi-agent Systems. 2000 IEEE, pp 3116-3121.
- [4] Korba, L, R, Liscano, a Distributed Framework for a Message System, MATA'99, Ottawa, Oct, 6-8, 1999.
- [5] F.J.Wang and S.Jusoh, Integrating multiple web-based geographic information systems, IEEE MultiMedia, pp

49-61, January-March 1999.

- [6] Kharola P.S., Gopalkrishna B. and Prakash D.C., 2000, "Fleet Management using GPS and GIS", Bangalore Metropolitan Transport Corporation (BMTTC) case study, *Map India 2001*.
- [7] Open GIS Consortium Web Map Servr Interfaces Implementation Specification, Revision 1.0.0. <http://www.opengis.org/specs/00-028.pdf>. September 20, 2000.
- [8] S.H.Wong, S.L.Swartz, A Middleware Architecture for Open and Interoperable GISs, IEEE MultiMedia, pp 62-76, April-June 2002.
- [9] F.C.Lin and J.Y.Hsu, Cooperation and Deadlock-Handling for an Object-Sorting Task in a Multi-agent Robotic System, Proc. Of IEEE Inter. Conf. On Robotics and Automation, Nagoya, Japan, May 1995, pp.2580-2585.

Min Peng is a Ph. D and lecturer of College of Computer, Wuhan University, China. Her current research interests include distributed computing, GPS/GIS application and multi-media data mining.

Efficient and Adaptive Load Balancing Based on Mobile Agent *

Yang Yongjian Chen Yajun Cao Xiaodong Ju Jiubin
College of Computer Science and Technology, Jilin University

Key Laboratory of Computer Communication of Ministry of Information Industry
Changchun, Jilin, 130012, P.R.China

Email: yyj@jlu.edu.cn, chenyajun@sohu.com, cxd1977cn@yahoo.com.cn, jjb@jlu.edu.cn

Tel: 13926998181; 0756 7855808

ABSTRACT

In this paper, firstly, we analyze some problems in the traditional load balancing, such as the structure, collecting and updating load information, adjusting strategy, and the extensibility. Secondly, we propose EALBMA (Efficient and Adaptive Load Balancing based on Mobile Agent) and discuss its basic principles. Using mobile agent, which is intelligent and mobile, EALBMA can resolve these problems above well. Therefore, EALBMA can improve the performance, adaptability, and extensibility greatly. Finally, we draw the conclusion that it is reasonable and necessary to improve load balancing using mobile agent.

Keywords: EALBMA, Load Balancing, Mobile Agent, Adjust Strategy, Performance, and Adaptability.

1. INTRODUCTION

With the rapid development of Internet and increase of users' need to network, LB (load balancing) has been applied extensively [1]. According to the controlling manner, LB can be classified into CC (centralized controlling) and DC (distributed controlling). However, there are some problems in both manners.

Table 1 describes the advantages and disadvantages of CC and DC. It is necessary to adjust the structure to resolve these problems. Moreover, there are common problems in both manners as follows:

- (1) In majority of LB systems, there is always one or few LB strategies, so they can't adapt themselves to different environments. It is almost impossible to add new strategies while system is running.
- (2) It is very difficult to adjust the current architecture and strategy to acquire better performance, after the structure is changed.

Table 1. The comparison of CC and DC

	Advantage	Disadvantage
CC	Exactly; High efficiency; Whole balance is easy;	Have bottleneck; Low reliability;
DC	No bottleneck; High reliability; Local balance is easy;	Message to interact is excessive; Low efficiency;

From the discussion above, we can conclude that there are

some problems in structure, reliability, performance, adaptability, and extensibility of load balancing at present. However, these problems are difficult to be resolved using traditional technology. It is necessary to seek a new way. After studying and analyzing mobile agent, we found that mobile agent can resolve these problems well [2].

Mobile agent (MA) is a novel technology originated from distributed network and artificial intelligence [3]. It has been applied into network management, electronic commerce and many other fields. Differing from traditional agents, MA can move among nodes in network. MA can take data and codes, which can be executed in destination nodes. MA can complete the corresponding tasks according to predefined rules and knowledge accumulated by itself.

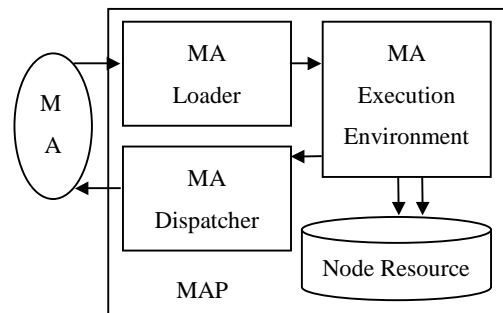


Fig.1 MA and MAP.

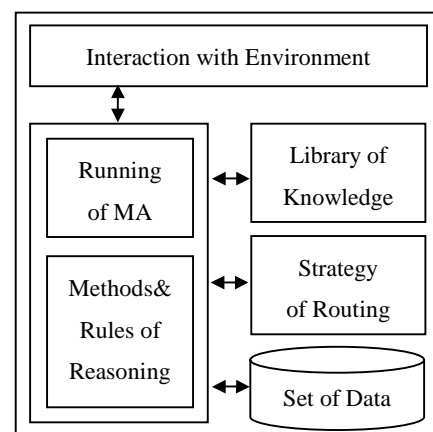


Fig.2. The structure of MA.

MA needs to run on MAP (Mobile Agent Platform) that works on operating system of nodes, as Fig.1. The structure of MA is described as Fig.2.

MA is very flexible and convenient. For example, in Aglets

* This research is supported by STPP (Science and Technology Planning Projects) [PC200320007] of Zhuhai, P.R.China.

[4], a MAP developed by IBM, we can create, clone, dispatch, retract, activate, and destroy MA. A MA in Aglets is a thread that needs few resources during running.

These characteristics of MA make it reasonable to improve LB as follows:

- (1) MA can reduce transmitting of data, save bandwidth and overcome network latency, because MA can move independent and transfer computation into data fields. So MA can improve the efficiency and performance of LB.
- (2) MA can improve the reliability of LB, because MA can be executed asynchronously, and it is robust and fault-tolerant.
- (3) MA is intelligent, mobile, flexible, and active, so it can substitute host to complete assigned tasks. By operating on destination nodes, MA can adjust these nodes. MA can also apperceive the change of environment and respond to it. Therefore, MA can be used to improve the adaptability and extensibility of LB [5].

This paper proposes a load balancing framework called Efficient and Adaptive Load Balancing based on Mobile Agent (EALBMA) that can resolve the problems we discussed above with the aid of mobile agent. The rest of the paper is organized as follows: Section 2 presents the structure of EALBMA. Section 3 and section 4 discuss the basic principles of EALBMA. Section 5 concludes this paper and put forward our future work.

2. THE STRUCTURE OF EALBMA

In section 1, we analyzed the disadvantages of CC and DC. In this paper, EALBMA resolves them as follows.

To avoid the bottleneck of CC, DC is adopted by EALBMA, and then every node receives tasks independently. To improve efficiency, adaptability and extensibility of system, a CN (control node) is used. It can collect and update the LI (load information), and monitor the running state of system. It would adjust the strategy and structure of system if necessary. These operations are all completed through MA. In this way, extra cost to balance load is low, and the performance, adaptability and extensibility can be improved greatly.

This way may result in a new bottleneck in CN. However it can be overcome by MA. MA can distribute computation into all nodes, and a MA can accomplish the work in one time that needs many interactions among nodes in traditional ways. This problem will be still discussed in the rest of this paper. Fig.3 illustrates the structure of EALBMA.

In this structure, nodes can be classified into SN (Serving Node) fixed serving module to serve for tasks, and CN (Controlling Node) fixed controlling module to control the system. CN is the best node in the system. It can also receive and execute tasks. But to improve its reliability, fewer tasks should be allocated to it.

Because the controlling to system is completed by MA in EALBMA, the CN wouldn't be the bottleneck. Moreover, EALBMA can easily settle the trouble of CN as follows: CN chooses a standby node and fixes the controlling module beforehand. If the trouble appears in current CN, the standby node will act as a new CN, and then control the system to keep

on working normally.

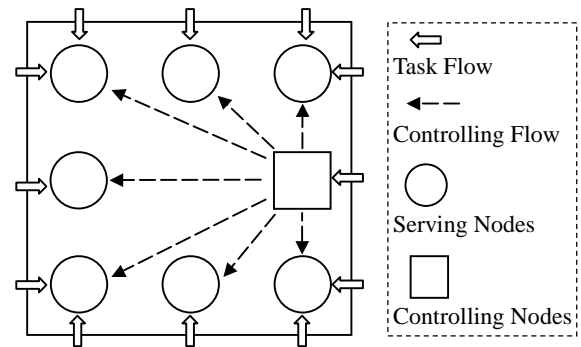


Fig.3. The structure of EALBMA.

3. COLLECTION OF LOAD INFORMATION IN EALBMA

To achieve LB, LI (load information) of each node should be collected first of all. In traditional methods, LI of every node is collected by local fixed agents of this node.

Traditional methods act as followings. In CC, each node sends LI to a load balancer, and then the balancer sends LI of all nodes to each node. In DC, every node broadcasts LI of itself to all the other nodes. However, both methods will consume plenty of bandwidth, and the former method will result in a bottleneck.

In this paper, EALBMA presents the following methods to collect and update LI.

Firstly, LIC (Load Information Collector) in every node collects current LI of local node every T_{info} time, and stores it in the LIL (Load Information Library). The value of T_{info} can be changed. For example, T_{info} can be increased (decreased) when the change of the load is quick (slow).

Secondly, MACLI (Mobile Agent for Collecting Load Information) dispatched by CN collects LI of every node in a cycle. When the MACLI reaches one node, it loads LI of this node. At the same time, it releases the LI of other nodes stored in MACLI to this node. This information released to current node is up to date because it is collected by MACLI just now. Additionally, if LI is almost not changed compared with the last time, MACLI will not load the current LI in order to save time and resource.

Finally, when the operations in one node is completed, MACLI moves to next node to do the same work according to predefined routing rules and current state of network. After a cycle, MACLI returns back to CN, and stores this information in CN. This information will be used to analyze by CN to adjust balancing strategy (See section 4). After this, MACLI starts next cycle. Fig.4 shows the principle of the collection of LI by mobile agent.

In the discussion above, the number of MACLI (N_{MACLI}) dispatched by CN in one time is decided by the number of nodes and the structure of system. In small system, one MACLI can achieve the task. But if there were many nodes in the system, N_{MACLI} would be increased. In this condition, CN can dispatch several MA (respectively named $MACLI_i$,

$i=1, \dots, N_{MACLI}$), and one MACLI will complete the LI collection of a set of related nodes. If the time of the cycle of $MACLI_i$ is T_{MACLI_i} , then the following two equations are needed to be accorded:

$$T_{MACLI_i} = T_{MACLI_j} = T_{MACLI} \quad (i, j = 1, \dots, N_{MACLI}) \quad Eq. (1)$$

$$T_{MACLI} = T_{info} \quad Eq. (2)$$

Eq.(1) can be beneficial to the synchronization and cooperation of multi-MACLI. Eq.(2) can ensure that LI collected every time is efficient and timely.

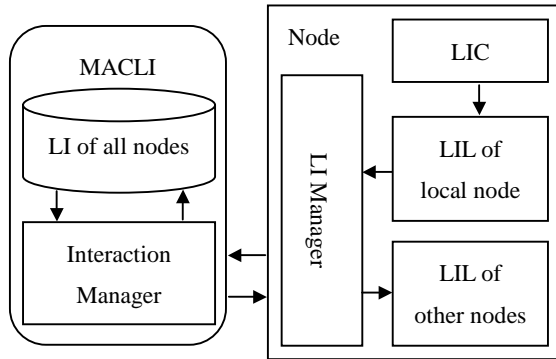


Fig.4. Load information collection on nodes.

4. STRATEGY OF LOAD BALANCING IN EALBMA

After collecting LI, EALBMA can balance the load according to current strategy. However there are many load-balancing strategies can be chosen. This section will introduce briefly Sender-Initiated Strategy Based on Threshold (SISBT) before discussing adjusting strategies in EALBMA.

4.1 SISBT Strategy of Load Balancing

In this subsection, we will describe SISBT in following three factors.

4.1.1 Resource Utilization Ratio

Resource utilization ratio of every node is the percentage of resource used in all resource can be provided of local node.

$$NV_{Use} = NV_{Current} / NV_{Total} \quad Eq. (3)$$

In Eq.(3), NV_{Use} is the resource utilization ratio between 0 and 1. $NV_{Current}$ is the being used resource now in local node. It can be computed according to Eq.(4).

$$NV_{Current} = \sum NW_i \times NV_i \quad Eq. (4)$$

NV_i is resource i being used in local node now. The resource can be CPU, Disk, Memory, I/O and others according to the requirement of tasks. LIC collects the LI about these resources using the relevant methods fixed in LIC, and computes the utilization percent of them. Finally, LIC stores this information in the LIL (Load Information Library).

NW_i is the weight value of resource i . NW_i is decided according to the importance of resource i to tasks, and can be adjusted while running. NV_{Total} is the total resource can be

used, which is fixed on every node. So NV_{Total} can be computed while the startup of system.

4.1.2 Threshold Values, Node State and Queues

Define two threshold values, which are L_H and L_L . They must accord with the condition: $1 \geq L_H > L_L \geq 0$. Their initial values are 0.9 and 0.1, but can be changed dynamically if necessary. Define three node states (NS) as follows:

- (1) **Heavy** ($NV_{Use} \geq L_H$): In this state, the node is busy, and can't receive new tasks and must allocate them to another node, whose NS is Normal or Light.
- (2) **Normal** ($L_H > NV_{Use} > L_L$): In this state, the node is normal, and can receive tasks from other nodes, whose NS are Normal or Heavy. It can also send tasks to another, whose NS is Normal or Light.
- (3) **Light** ($L_L \geq NV_{Use}$): In this state, the load of node is light. It can receive tasks from other nodes, whose NS are Normal or Light.

Every node arranges other nodes into three queues in local node. According to the state, these nodes can be divided into **Queue_Heavy**, **Queue_Normal**, and **Queue_Light**. They are ordered by NV_{Use} from small to large respectively.

4.1.3 Algorithm of Allocating Tasks

This paper introduces the following algorithm of allocating task in SISBT.

```

Receiving a new task TA;
IF (NS=Light)
    execute(TA, LocalNode);
IF (NS=Normal)
    IF (Queue_Light=null)
        IF (Queue_Normal=null)
            execute(TA, LocalNode);
        ELSE IF (UseOfRemote(TA, first(Queue_Normal))
            << UseOfLocal(TA, LocalNode))
            execute(TA, first(Queue_Normal));
        ELSE execute(TA, LocalNode);
    ELSE execute(TA, first(Queue_Light));
IF (NS=Heavy)
    IF (Queue_Light=null)
        IF (Queue_Normal=null)
        {
            Report failure to control node;
            execute(TA, LocalNode);
        }
    ELSE IF
        (UseOfRemote(TA, first(Queue_Normal))
        << UseOfLocal(TA, LocalNode))
        execute(TA, first(Queue_Normal));
    ELSE execute(TA, LocalNode);
    ELSE execute(TA, first(Queue_Normal));

```

In this paper, in order to be brief, we only consider allocating new tasks. However the running tasks can be also allocated in EALBMA if necessary.

4.2 Adjusting of Strategy

In EALBMA, the adjusting of strategy is classified into next three levels: (1) Adjusting parameters. (2) Switching in existent strategies. (3) Adding new strategies. They can be applied into different situations.

Mobile agent can realize adjusting of strategy above easier than traditional methods, because mobile agent is intelligent and mobile. Moreover, the asynchronous quality and dynamical routing specialty of MA can improve the reliability of system.

4.2.1 Adjusting Parameters

In section 4.1, we defined many parameters. If needed, we can adjust them. MACLI stores LI of all nodes in CN after a cycle. Then CN can analyze every NV_{Use} and whole load of all nodes, and can decide whether it is necessary to adjust parameters or not. The adjusting can be achieved as follows:

- (1) If load changes quicker (slower), T_{info} should be smaller (bigger).
- (2) If whole load is heavier (lighter), the threshold values, L_H and L_L , should be higher (lower).
- (3) If one kind of resource is more (less) important to tasks while being executed, its weight value should be bigger (smaller). And so on.

If deciding to adjust some parameters, CN will dispatch a MAAS (Mobile Agent for Adjusting Strategy) to adjust the parameters to necessary nodes. The adjusting of mobile agent to one node likes Fig.5.

4.2.2 Switching in Existent Strategies

While designing a strategy of load balancing, we need to set up corresponding models and design appropriate strategies, according to the structure and tasks of system. No one strategy can satisfy all situations, and different situations need related strategies. So, to improve the adaptability of system, we must adjust the strategy itself (i.e. replace current strategy with another) other than adjusting parameters. This section introduces the switching in existent strategies and section 4.2.3 will describe how to add new strategies into system. It is not frequent to switch in existent strategies. Only when one or two of following two conditions appears, it is just necessary. Firstly, there is a serious unbalance in the whole system. Secondly, the failure of first allocation of tasks is frequent.

CN analyzes the running situation from LI in latest period of all nodes. MA dispatched by CN collects the information and send it to CN. The information includes:

- (1) Requirement to resource of tasks;
- (2) Resource utilization ratio of every node (NV_{Use});
- (3) The ratio executing tasks in local node;
- (4) The ratio executing tasks in a remote node after allocating tasks;
- (5) The ratio allocating tasks more than once.

The mobile agent can also analyze some information during collecting and sending back this information, which can light the burden of CN by distributing some computation into the other nodes.

During analyzing strategies, EALBMA adopts the method of simulation and analysis. However we can use other methods, such as expert system and nerve network, to achieve it, too.

In our method, CN stores some strategies in strategies library, such as, least-connections, round robin, send-initiated, receiver-initiated, find-best, find-first, and so on. When necessary, CN simulates tasks in latest period with the

strategies above respectively, and analyzes and compares the execution result of them. If finding a strategy is superior to current one obviously, CN will decide to replace the current by this. If not, CN will record this event and report it to the administrator to quest better one. In addition, the administrator can also choose strategy manually through experience. If deciding to replace, CN dispatch a mobile agent, which moves to every node to activate and start the chosen strategy from strategies library. Fig.5 describes the adjusting of mobile agent to one node.

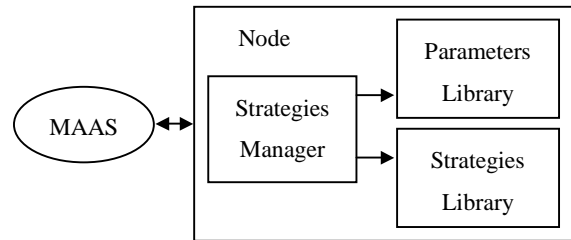


Fig.5. Adjust strategies using mobile agent.

4.2.3 Adding New Strategies

The adjusting of strategy discussed in section 4.2.2 is replacing current strategy with existent and chosen one in strategies library. However, sometime, if developers develop some new strategies fit current system, EALBMA can add them into system while system is running, and can't affect the working. MAAS takes these new strategies and move to every node. Then load and fix them into each node. Thus, the new strategies can be activated and used when necessary.

5. EXTENSIBILITY OF EALBMA

In traditional load balancing, once the structure of system is confirmed, it is difficult to adjust the structure and controlling strategies according to the change of system (increasing or decreasing nodes, and changing physical location of nodes, etc.).

However the extensibility of system in EALBMA is better. Generally, we use one CN to control the system. But if the nodes are excessive, one CN can't control the system well. So it is necessary to increase CN. We can transfer the controlling structure of two layers into three or more layers. That is to say, the nodes, which are all controlled by one CN in one region before, can be controlled by multiple CN in different regions respectively.

For example, originally there are nodes A_1-A_n controlled by $U_{control}$, and then B_1-B_m are added later. If system continues to use original controlling structure and controlling mode, the whole performance will be affected. In EALBMA, after the computation and analysis by $U_{control}$, $U_{control}$ makes the following decision: Choose one node ($A_{control}$) from A_1-A_n to control A_1-A_n . Choose one node ($B_{control}$) from B_1-B_m to control B_1-B_m . At the same time, $A_{control}$ and $B_{control}$ is control by $U_{control}$. Fig.6 illustrates this process.

The process above is achieved according to following method: $U_{control}$ dispatches mobile agent to fix and setup controlling module into $A_{control}$ and $B_{control}$. The mobile agent also modifies the related parameters on all node included severing nodes and controlling nodes to optimize the performance of the system further. It is very difficult to complete these

operations using traditional method. But this adjusting is very rapid and intelligent using mobile agents in EALBMA.

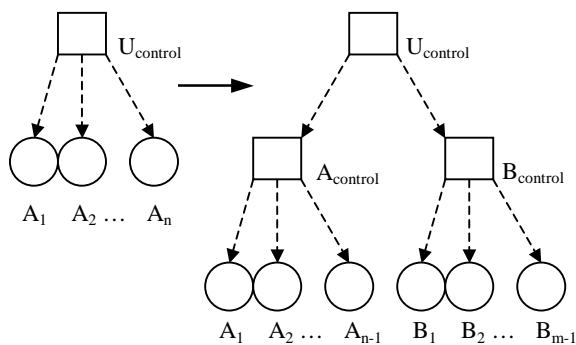


Fig.6. The change process of structure in EALBMA.

After the operations above, the load balancing can be processed in every region respectively in order to improve the efficiency. If there is obvious unbalance among the regions, $U_{control}$ can allocate tasks across regions or adjust the strategy of related regions. In this way, EALBMA achieves whole balance. In a word, the extensibility of EALBMA is excellent.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we can conclude that EALBMA can improve the performance, adaptability, and extensibility using mobile agent.

EALBMA can be applied into the load balancing on Web server [6] or distributed computation. It can be used in LAN and WAN. However it is more excellent in WAN because mobile agent can easily overcome the limitation of network bandwidth and latency in it [7]. Furthermore, being able to adjust strategies, EALBMA can be used to test and evaluate various strategies.

Now, we are trying to apply load prediction to load balancing using mobile agent [8], in order to improve the performance further. Our future work is to implement and test EALBMA scheme with load prediction.

7. REFERENCES

- [1] M.J.Zaki, W.Li, et al, "Customized Dynamic Load Balancing for a Network of Workstations", Proceedings of the 5th IEEE Int. Symp., HPDC, 1996, pp.282-291.
- [2] Jiannong Cao, Yudong Sun, et al, "Scalable load balancing on distributed web servers using mobile agents", Journal of Parallel and Distributed Computing, Vol.63, No.10, Oct. 2003, pp.996-1005.
- [3] V.A.Pham, A.Karmouch, "Mobile Software Agents: An Overview", IEEE Communications Magazine, Vol.36, No.7, July 1998, pp.26-37.
- [4] Aglets Software Development Kit, URL: http://www.trl.ibm.com/aglets/index_e.htm.
- [5] J.Gomoluch, M.Schroeder. "Information agents on the move: A survey on load balancing with mobile agents". Software Focus, 2(2), April 2001.
- [6] V.Cardellini, M.Colajanni, P.S.Yu, "Dynamic load

balancing on Web-server systems", IEEE Internet Computing, Vol.3, No.3, May/June 1999, pp.28-39.

- [7] R. Lueling, B. Monien, F. Ramme, "Load Balancing in Large Networks: A Comparative Study", Proceedings of the 3rd IEEE Symposium on Parallel and Distributed Processing, 1991, pp.686-689.
- [8] Wei JIE, Wentong CAI, Stephen J. TURNER, "Dynamic Load-Balancing Using Prediction in a Parallel Object-oriented System", International Conference on Parallel and Distributed Systems (ICPADS) 2001, pp.279-288.



Yang Yongjian is a Full Professor of Computer College, the vice Dean of Software College in Jilin University. He is also a Committeeman of Directing Committee for Computer Teaching of National Education Ministry, the Director of Key Laboratory of Computer Communications of Ministry of Information Industry, and a syndic of Computer Academy of Jilin Province. He received the master degree of Computer Application from Beijing University of Posts and Telecommunications in 1991. Since then, he has been studying Computer Network Communications. He has presided more than 10 important projects, and published one book and over 30 journal papers.



Chen Yajun is a graduate student, majoring in Distributed Network under the instruction of Yongjian Yang professor at present. After received the bachelor degree of Computer Communications in Jilin University in 2002, he was recommended to Computer College in Jilin University without graduate entrance examination. His research interests are in Load Balancing, Load Prediction, Mobile Agent and Active Network. He has attended 3 important projects and published 2 journal papers.

A Mobile Agent Based Middleware for Grid Computing

Peng Dewei, He Yanxiang

School of Computer, National Key lab of Software Engineering,

Wuhan University, Wuhan, Hubei 430072, China

Email: pdw512@tom.com Tel: 86-27-87668957

Email: yxhe@whu.edu.cn Tel: 86-27-87642413

ABSTRACT

Grid provides computational power beyond the capacity of even the largest parallel computer system, and merges extremely heterogeneous physical resources into a single virtual resource. In spite of its ambitious goals, grid computing still benefits only a handful of researchers who are free to forage among supercomputers and high-performance workstations for their computation-intensive projects. The majorities of users, on the other hand, have few opportunities to access such computing facilities and are assumed to pay little attention to computational grids in favor of their own desktop computing environments. If they could authorize each other to mutually use their computers, a collection of such desktop machines would consistently provide them with an enormous computing resource. In this type of grid systems, besides performance, adaptive and platform independence is the key issue. This paper describes the architecture of a mobile agent based middleware for grid computing environment, whose aim is acquire adaptive and high performance computing power, and explores the benefits of using mobile agent as its key middleware building technology. We describe the adaptive mode of task execution of the system, as well as our current work and future plan.

Keywords: mobile agent, grid computing, middleware, broker mode.

1. INTRODUCTION

Grids are geographically distributed platforms for computation, accessible to their users via a single interface. They provide computational power beyond the capacity of even the largest parallel computer system, and merge extremely heterogeneous physical resources into a single virtual resource. While there is considerable variation in what is meant by the term "grid" [1]. Current grid computing research concentrates primarily on the problem of connecting large supercomputing sites to create computing environments for solving very large scientific problems. These systems will run relatively few programs submitted by few sophisticated users. In spite of its ambitious goals, grid computing still benefits only a handful of researchers who are free to forage among supercomputers and high performance workstations for their computation-intensive projects. The majority of users, on the other hand, have few opportunities to access such computing facilities and are assumed to pay little attention to computational grids in favor of their own desktop computing environments [2][3]. However, if they could authorize each other to mutually use their computers, a collection of such desktop machines would consistently provide them with an enormous computing resource, we believe that supercomputer-based grids couldn't be the mainstream in future grids, because it can't satisfy the requirements of different users from current research.

Grid computing system used by most common users could be characterized by a very large set of services offered to a very large user population, where users submit many computing tasks for execution. The tasks are probably sequential or parallel. These users will not tolerate long queuing delays, since most jobs will be subtasks of interactive applications; instant processing is required. They also want an adaptive way to execute their tasks.

In this type of grid systems, besides performance, adaptive and platform independence is the key issue. At the moment, Java is the best candidate language to achieve this. The goal of our project is to create a mobile agent based middleware for a large-scale, global computing environment to acquire adaptive and high performance computing power, and explores the benefits of using mobile agent as its key middleware building technology.

2. RELATED WORK

There are several ongoing grid-related projects that use mobile agent as its middleware. This section differentiates our system from other middleware systems based on mobile agents for grid computing. NetSolve is a well-known system that uses an agent-based approach for resource allocation [4]. The system provides user programs with an RPC environment. The main difference with our proposed middleware is that a NetSolve agent is local to one site and intended to orchestrate each application's RPCs over the network, while we use mobile agents to dispatch an entire job. MASA is a proposed system that allows those involved in IT management to dispatch their mobile agents to a target machine where the agents refer to a site-specific access list, authorize each others, and complete a software installation cooperatively [5]. In MASA, mobile agents representing different users perform a task at a specific site cooperatively. On the other hand, each mobile agent in our middleware searches available sites and launches a parallel or sequential job according to the workload. Ref [6] [7] [8] are mobile-agent-based systems proposed for computational grids. The most similar project to ours is JM [3], a framework to build global computing for sequential or parallel tasks to execute. However, this system is based on jini [9] which is the key technology. Our system is different from these in that it is a pure Java system with mobile agent to execute tasks. These systems' design principle overlaps with our proposed middleware in terms of using mobile agents; however the originality of our approach lies in having each mobile agent independently and entirely to take care of a different client job from resource search to job migration. We are more concerned with job execution mode in order to acquire more adaptive computing mode.

In the next section we briefly mention related grid projects that use mobile agent. In Section 3 we introduce the

architecture of our system, and in Sections 4 we describe the supported sequential and parallel tasks. Section 5 describes work currently in progress, and the paper ends with our conclusions and discussion of future plans.

3. SYSTEM ARCHITECTURE

The MAM, as shown in Figure-1, is essentially a mobile agent-based middleware for grid computing. The broker executes trades between clients and servers and forms a grid computing environment upon receiving an agent task. The agent is then cloned for each server. The agent tasks are run sequentially or parallel. They are executed on the host independently of the broker. Servers of the machine can report results to the broker or directly to clients. Notice that the system may comprise of more than one broker. Each broker serves regional clients and servers or nationwide domain-specific clients and servers. Brokers are organized in a hierarchical way for a wide-area computational grid.

The middleware relies on mobile agent technologies to realize ubiquitous global computing. Its implementation is based on Java's remote method invocation (RMI) and object serialization. The RMI system allows remote-procedure-call (RPC) like access to remote objects and supports mobile behaviors [10][11].

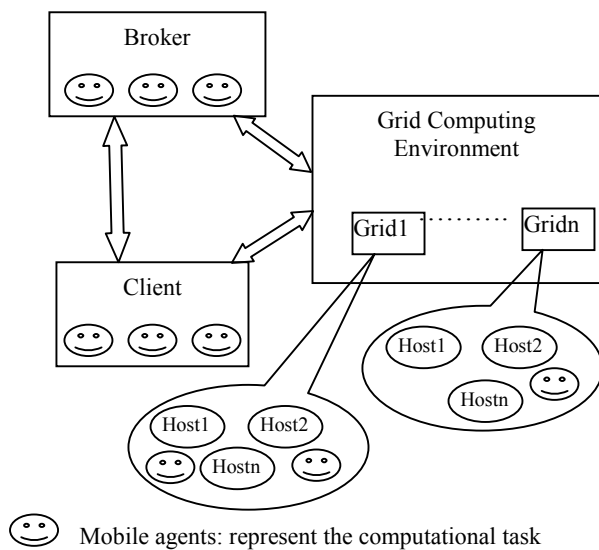


Fig-1 Architecture of Mobile Agent Based Middleware for Grid Computing

Like traditional RPC, RMI is enabled by declaring a remote interface of an object to expose its methods to remote objects. The following **AgentHost** interface enables clients to create mobile agents and dispatch them to a Host that implements the interface:

```
import java.rmi.*;
public interface AgentHost extends Remote
{
    Object execute( AgentTask agenttask ) throws
        RemoteException;
}
```

The computation of grid systems will face many problems

such as severe network delays, network errors and service going down etc. Mobile agent can represent the client to execute the task in remote host, even the client is off-line during the execution, the result can be returned after the client is reconnected. This is one of the merits of the mobile agent used for grid computing.

Our work concentrates on providing adaptive computing mode. We present four execution models relied on mobile agent. These agents are used to return the final results to the client. The detail of the execution models will be discussed in the section 4.

3.1 The Mobile Agent

It is assumed that in this computational environment clients send computing tasks as mobile agents to computing services. In our system, every client task must implement the **AgentTask** interface.

```
public interface AgentTask extends Serializable {
    public initialize();
    public execute(object broker);
    public terminate();
}
```

The **AgentTask** interface declares three primary abstract methods **initiate()**, **execute(object broker)**, and **terminate()**. The **initiate** method creates an Agent object. The **execute(object broker)** will be invoked by the compute service when the task reaches its final destination. When an Agent spawns other surrogates, it uses the broker (located by the broker parameter) to deal with the problems of service locating and task scheduling. The **terminate** method stops the execution of the Agent object for either transfer, or storage, or termination. It suspends each of its threads on the host

3.2 The AgentHost service

A basic implementation of the **AgentHost** is as follows. The execute method takes the agent object and starts the execution of the object. **RMI** provides for secure channels including encrypted sockets between client and server. It also uses built-in Java security mechanisms to protect servers from possible attacks by un-trusted clients. It is realized by installing a security manager before exporting any server object or invoking any method on a server. RMI provides the **RMISecurityManager** type that is as restrictive as those used for applets. Each server can also define and install its security manager object to enforce different security constraints. For example, a server can open a tmp directory for an agent to store intermediate results. The server should also allow alien agents to open connections to the broker and the hosts.

```
import java.rmi.*;
import java.rmi.server.*;
public class AgentHostImpl implements AgentHost
{
    public AgentHostImpl() throws RemoteException {}
    public Object execute(AgentTask agenttask) throws
        RemoteException;
    {
        return agenttask.execute(object broker);
    }
    public static void main(String args[])
    {
        System.setSecurityManager( new
            RMISecurityManager() );
    }
}
```

```

Try {
    AgentHost as = new AgentHostImpl();
    Naming.rebind( "AgentaHost", as );
} catch ( IOException ioe){};
}

```

3.3 The Broker Service

Figure-1 shows a mobile agent-based middleware for grid computing. The broker executes trades between clients and servers and forms a grid computing environment upon receiving an agent task. The agent is then cloned for each server. Notice that the system may comprise of more than one broker. Each broker serves regional clients and servers or nationwide domain-specific clients and servers. Brokers are organized in a hierarchical way for a wide-area computational grid. There is only one broker in our current prototype, we will add more in our future work.

The role of the Broker is to manage the computing services, allocate mobile agents, and interface clients with grid. We assume that grid computing systems would be very frequently accessed. Therefore it is advantageous to make use of broker based on mobile agents to collect and keep track of specified computing service.

The broker constructs a virtual machine based on the workload information of registered servers. The information can be either polled by the broker or reported by the servers. Information polling is realized by a normal information-collection agent. It is dispatched by the broker and is migrated from server to server to collect the workload information of servers periodically. Since there is no agreement about a single metric for server workload in the literature, the agent-based information collector provides the broker with a way to define customized services (workload indices) from the servers.

Specifically, a client defines a computational task as an **AgentTask**. The **GetHost()** method returns the matching computing services. The broker receives the agent by the method **ReceiveAgent**. Then it sends them to the matching computing service by the method **SendAgent**. If the client does not want to deal with task scheduling, the complete agent task can be delegated to the broker by the method **ForwardAgent**. Then, the broker will select the best matching service for execution. In our current execution, if multiple matching services are found, the one with the lowest number of active threads is selected. We adopt the algorithms presented by Ref[15] to provide effective and efficient task scheduling and workload balancing. In our future plan, more sophisticated algorithms will be used.

```

public interface AgentBroker {
    public HostList GetHost(ComputingService service)
        throws RemoteException;
    public Object ReceiveAgent(AgentTask agenttask);
    public Object SendAgent(AgentTask t, HostList host)
        throws RemoteException;
    public Object ForwardAgent(AgentTask t, HostList host)
        throws RemoteException;
}

```

4. EXECUTION MODE OF THE TASK

All interactions among clients, grid, and broker are coordinated by the mobile agent, which handles the execution of the task and the reception of results and exceptions, in this section we present four different modes of task execution based on mobile agents for grid computing system which are called H-Broker mode, F-Broker mode, P_Broker mode, Enhanced P-Broker mode.

4.1 H-Broker Mode

The simplest of all modes is the handle-driven broker mode. Fig-2 shows the call sequence of the key operations of the program in the H-ORB. How does the client request the service? When a service is requested, the client performs a request operation with the broker which returns the handle for the desired host. In this paper the client always requests the agent, even though it has called the same server and saves the address of the server. This ensures that the request is routed to an active server in case the server used in the previous cycle is invalid. The client uses the handle to send the mobile agent to the specific server and the mobile agent returns the result to the client. Depending on the task of the mobile agent and architecture of middleware, the client may block waiting for the reply or proceed to do some other tasks and receive the reply later. We have designed the client to wait until the reply is received. It works as follows: client sends the request to the agent the agent sends the handle-driven of the server to the client client send the mobile agent to the specific sever the mobile agent returns the result.

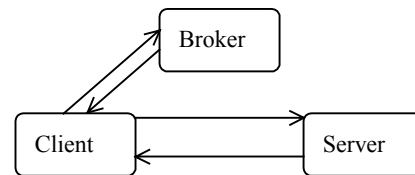


Fig-2 H-Broker mode

The broker is used only to retrieve one or more suitable compute services, then, the client uses these services directly. One of the advantages of using a middleware in the grid system is that it can collect compute service proxies from many Grid services, hence can speed up the retrieve process and save time for the client. We illustrate this execution mode with the classic θ computation program. This program is designed for parallel execution though in this case only one processor will be used, so we use the formula

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + K + \frac{x^n}{n!} (x=1) \quad \text{but not the}$$

formula $e = (1 + \frac{1}{n})^n (n \rightarrow \infty)$ to compute θ . The

detailed algorithm is illustrated as below:

```

public class EComputing implements AgentTask {
    protected int beg //initial value of i
    protected int num; //end value of i
    protected double evalule; //the result
    ...
    public Object execute(Object broker) ... {
        double sum = 0.0;
        if (beg = 0) {
            sum = 1.0;
        }
    }
}

```

```

for (int i = beg; i <= num; i += 1) {
    sum += 1/factorial(n);
    // factorial(n) is the method to compute the factorial of n
}
evaluate = sum;
return new Double(evaluate);
}
}

```

The broker uses method **GetHost(Computing Service service)** to retrieve one or more suitable compute services, then, the client uses these services directly. In this program an Agent of **EComputing** will be sent to the host from the client.

4.2 F-Broker Mode

The call sequence of the key operations of the program in the F-Broker mode is presented in Fig-3. It is obvious that the total number of messages per server interaction is lower than the F-Broker mode. The client sends the execution of its tasks to a broker, the broker then forwards the received task to a selected compute host with the service request. The server performs the mobile code (requested service) and mobile agent returns the results back directly to the client. It works as follows: client sends the task to the broker the broker forwards mobile agent to the specific sever with the service request the mobile agent returns the result.

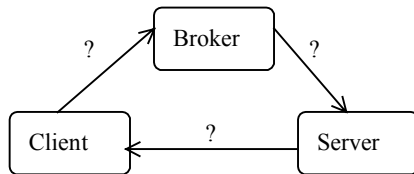


Fig-3 F-Broker mode

```

EComputing ec
try {
    evaluate = ec.execute(broker);
} catch (Exception e) {...}

```

F-Broker mode simplifies client code and reduces the resource requirements of the client as well as the brokered sequential execution described in ref[3]. Ref[3] use asynchronous communication mode to solve the severe network delay, but its broker is the neck bottle of the system under the mode of brokered sequential execution. We use the mobile agent to avoid this problem. The client and the broker needn't to wait until the mobile agent comes back with the results. It invokes the method of **execute(broker)** to send task to the broker, which forwards the received task to a selected compute host. The broker gets the appropriate compute host list by the method **GetHost**. Then the task received by the method **ReceiveAgent** is forwarded to the selected compute host by the method **ForwardAgent**. The call is now made the following way.

```

try {
    broker.GetHost(ComputingService service);
    broker.ReceiveAgent(AgentTask agenttask);
    broker.SendAgent(AgentTask t, HostList host)
} catch (Exception ex) {...}

```

We also can use the method **SendAgent** to forward the task. The difference between them destination the result is returned.

If you use the first method, the result will be returned to the client directly. If you use the second one, the result will be returned to the broker. This mode is also discussed in the paper.

4.3 P-Broker Mode

The mode of the P-Broker is similar to the mode of H-Broker. The main difference between them is that the P-Broker invokes the multi-server concurrently. In Fig-4, the client delegates the tasks of execution to the broker. The broker divides the task into subtasks. These tasks are allocated to compute hosts by the broker, who will also collect and return the final result. In this paper, we will assume all the tasks can be requested independently. Thus, all of subtasks can be executed concurrently by mobile agent. After completing computing, each host sends its reply back to the broker. The broker combines the results and sends a single reply back to the client. This mode can be used in support the execution of parallel programs using different execution paradigms. Without sending back the result, it can be used in remote computer configuration, network management, etc. That is out of the scope of this paper. We illustrate parallel program execution using the previous example. Several mobile agent need to be started to compute partial results.

```

try {
    broker.GetHost(ComputingService service);
    broker.ReceiveAgent(AgentTask agenttask);
    for (i=1; i<subtskno; i++){
        EComputing t[i] = new EComputing(beg,num);
    }
    broker.SendAgent(t[], HostList host)
} catch (Exception ex) {...}

```

The **SendAgent** method should send the subtask to the remote compute hosts. The result will be returned to the broker. The partial results can be combined in the program using the method **CombineResult()**. Then, the broker sends a single reply back to the client. The execution is illustrated in Fig- 4.

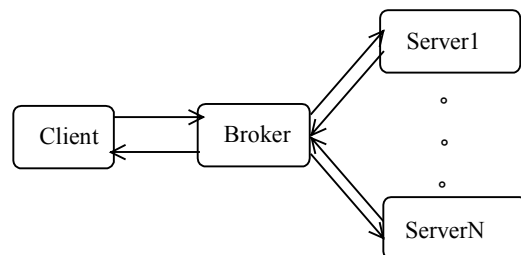


Fig-4 P-Broker Mode

```

//the final result
double evaluate = 0.0;
broker.ReceiveAgent(AgentTask agenttask);
try {
    for (int i = 0; i < subtskno; i++) {
        Double result = t[i].getResult();
        evaluate += result.doubleValue();
    } catch (Exception ex) {...}
    return new Double(evaluate);
}

```

In Fig-4, the client delegates the tasks of execution to the broker. The broker divides the task into subtasks and sends them to the matching computing hosts. The execution can be either synchronous or asynchronous.

Maybe the subtask $t[i]$ is too complex to the remote host, so it is necessary to divide the subtask to sub-subtasks. To acquire adaptive mode and fine granularity of the task execution, we present another mode of task execution called Enhanced P-Broker mode which is illustrated in Fig-5.

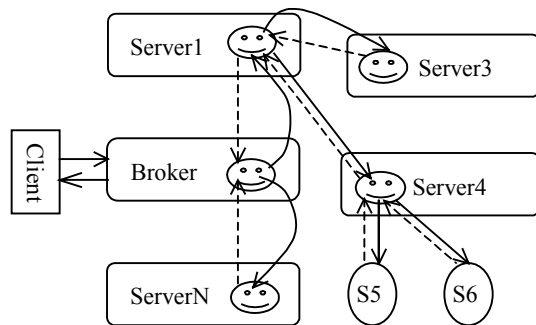


Fig-5 Enhanced P-Broker Mode

The task in the broker spawns two subtasks to compute the result respectively. If the subtask is above a threshold, the subtask will spawn two or more sub-subtasks to compute the result. These sub-subtasks are allocated by the broker, and they perform the same steps recursively; if necessary, they will start new subtask execution, such as mobile agent in server5 and server6 in fig-5.

5. WORK IN PROGRESS

Our system is still under development. Some features, such as sophisticated scheduling, security and authentication, load balancing are yet to be added. In this section we outline work that is currently in progress.

Several agent based load balancing solutions have shown very significant improvement in system performance in comparison with the system without load balancing method. We investigate the related work of load balancing and present three load balancing methods:

- **Random** is a very simple method. When the mobile agent comes to the broker, it takes task if there is something in the queue. In addition, it will randomly decide whether it will let a task to the node or not. It is obvious that this method does not use any information about nodes' load.
 - **Taskfewer-first** method uses collected data and makes decision according to the number the mobile agent in the hosts. If a task is sent to the broker, the broker searches a host which has the smallest number of mobile agent and sends or forwards the task to that host.
 - **Threshold** method continuously calculates average load for all host. If number of tasks on a host is bigger than average, some tasks will be taken from the node. On the contrary, if host has number of tasks smaller than average, some tasks will be sent to the node.
- We will try to investigate the performance of the middleware with the load balancing by the experiment and simulation

6. CONCLUSIONS AND FUTURE WORK

We have designed and implemented a middleware based on

mobile agent for grid computing system that aims to use a very large set of compute services for many smaller sequential or parallel tasks. The system provides various modes of task execution. Our tests and experiments show that mobile agent technology provides proper services and classes that can be used effectively with little modification to create large and robust computational grids.

As mentioned in ref[12], middleware based on mobile agent for grid computing system is one of the future research directions on mobile agent. However a number of research focus on solving the problem of interoperability using the middleware for mobile agent system but not for the grid computing system [12] [13] [14]. More research is needed on the performance analysis of middleware architecture based on mobile agent for grid computing systems. Our future work will primarily focus on refining the system for performance and demonstrating the feasibility of the system with more applications. Our work continues with adding security and user authentication functionality to the system and considering more factors such as security, serious network delay, mobile agent size, etc which will have impact on the performance of the middleware based on mobile agent.

7. ACKNOWLEDGEMENTS

The investigation is supported by Key science and technology development project of Wuhan City-"Research and application of e-Commerce intelligent analysis system". No: 20011007087.

8. REFERENCES

- [1] D.B. Skillicorn, "Motivating Computational Grids", Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02), pp:401-406
- [2] Munehiro Fukuda, Yuichiro Tanaka, Naoya Suzuki, Lubomir F. Bic, Shinya Kobayashi "A Mobile-Agent-Based PC Grid", Proceedings of the Autonomic Computing Workshop Fifth Annual International Workshop on Active Middleware Services (AMS'03), pp:142 -150.
- [3] Zoltan Juhasz, Arpad Andics, Szabolcs Pota1, "JM: A Jini Framework for Global Computing", Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02), pp:395-400
- [4] NetSolve Web Site. <http://icl.cs.utk.edu/netsolve/>.
- [5] G.Vogt. "Delegation of tasks and rights", Proceedings of the 12th Annual IFIP/IEEE Int'l Workshop on Distributed Systems: Operations & Management DSOM 2001, pages: 327-337, Nance, France, October 2001. INRIA.
- [6] W. Binder, G. D. M. Scrugendo, and J. Hulaas. "Towards a secure and efficient model for grid computing using mobile code", Proceedings of the 8th ECOOP Workshop on Mobile Object Systems: Agent Application and New Frontiers, Malaga, Spain, June 2002.
- [7] S.Hariri, M.Djunaedi, Y.Kim, R.P.Nellipudi, A.K.Rajagopalan, P.vdlamani, and Y.Zhang, "CATALINA: A smart application control and management environment". In 2nd Int'l Workshop on

- Active Middleware Services, 2000.
- [8] O.Tomarchio, L.Vita, and A.Puliafito, "Active monitoring in grid environments using mobile agent technology". In 2nd Int'l Workshop on Active Middleware Services, 2000.
 - [9] Jini Technology Core Platform Specification, <http://www.sun.com/jini/specs>.
 - [10] Sun Microsystems. Java remote method invocation—distributed computing for Java. <http://java.sun.com> (white paper).
 - [11] Rickard Oberg: Mastering RMI: Developing Enterprise Applications in Java and EJB (ISBN:0-471-38940-4), 2001, published by John Wiley & sons, Inc. pp:151-176
 - [12] D. Kotz, R. Gray, and D. Rus, "Future directions for mobile agent research". Technical Report TR2002-415, Department of Computer Science, Dartmouth College, Jan. 2002.
 - [13] P.Bellavista, A.Corradi and C.Stefanelli, "CORBA Solutions for Interoperability in Mobile Agent Environments" Distributed Objects and Applications, 2000. Proceedings of DOA'00. International Symposium on, 21-23 Sept. 2000, pp:283-292.
 - [14] P.Bellavista, A.Corradi and C.Stefanelli, "Middleware services for interoperability in open mobile agent systems", Microprocessors and Microsystems 25(2001), pp:75-83
 - [15] Sasa Desic, Darko Huljenic, "Agents Based Load Balancing with Component Distribution Capability", Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'02), pp:353-359.



Peng Dewei was born in 1976. He received the BE and ME degrees in Wuhan University, China, in 1998 and 2001, respectively. He is a currently PhD candidate in the Department of computer Science and Technology, Wuhan University, China. His Research interests include mobile agent, distributed computing and web systems.



He Yanking was born in Hubei, China, on January 1952. He received the BE degree in Computing Mathematics from Wuhan University, China, 1973 and ME degree in information systems from Oregon University, USA, 1986 and PhD degree in computer Science from Wuhan University, China, 1999. He academically visited Oregon University,

USA, in 1997 and 2001, respectively. He is currently a professor and PhD tutor of computer science and dean of National key lab of software engineering in Wuhan University, China. His research interests include multi- and mobile agent, distributed computing, software engineering, data mining. Professor He has published over 100 technical papers and is the author of twelve books. He also was awarded and National Special Prize by the Chinese Government in 1993.

Research on Prediction Model of Dynamic Load-Balancing with Mobile Agent in a Parallel Distributed System*

Yang Yongjian Cao Xiaodong Chen Yajun
College of Computer Science and Technology, Jilin University,
Changchun, CHINA 130021
Email: yyj@jlu.edu.cn Tel: 13926998181

ABSTRACT

In parallel-distributed system, the task distributed to every processing element is imbalance, and so the load of every PE is imbalance, so we introduce load balance mechanism. In this paper, by combined mobile agent and prediction mathematical model, we propose a simple and effective prediction algorithm of load balancing using in a parallel-distributed system, so the resource in system is fully utilized and processing rate of task is improved.

Keywords: Parallel Distributed System, Dynamic Load-balancing, Performance Prediction, Rate of Load balance Change

1. INTRODUCTION

In a multiprocessor distributed parallel system, load imbalance of every processing element (PEs) will appear when the task distributed to every processing element is non-uniform. We introduce load balance mechanism in this system for reasonably using the resource of every PE. Maybe the structure of every PEs is dissimilar, but they are independent and symmetrical in logical structure. All tasks running on PE have equality, so they are not differentiated between primary and secondary. At the same time, these tasks are distributed as a whole.

There are a lot of relevant studies and works on load balancing technology in parallel-distributed systems. On the whole, load balancing technology can be divided into three classes from their property: static load balancing and dynamic load balancing, implicit load balancing and explicit load balancing, distributed load balancing and centralized load balancing. We can dispatch the task to every PE by static load balancing strategy established at the beginning for these tasks that we can know runtime load condition before running. But for unpredictable computing and communication, as well as unpredictable runtime state of task, we must make a dynamic decision by dynamic load balancing. Implicit load balancing refers to load balancing performed automatically by the system, whereas explicit means that it is up to the user to decide when and which tasks should be migrated. In centralized load balancing a single PE is responsible for maintaining global load information and for making load-balancing decisions. In distributed load balancing, decisions are made locally by each PE and load information is maintained across all PEs which share the responsibility of achieving global load balance. In this paper, we focus on dynamic, implicit,

distributed load balancing strategies.

Because there are a lot of distribute characters in this system, we can introduce Mobile Agent (MA) technology into dynamic load balancing model. MA is a set of autonomous, intelligent programs that move through a network, searching for and interacting with services on the user's behalf. These systems use specialized servers to interpret the agent's behavior and communicate with other servers. A MA has inherent navigational autonomy and can ask to be sent to some other nodes. MA has many characteristics: reduce network load, overcome network delay, asynchronous and autonomous execution, dynamic adaptation, and so on. For dynamic load balancing strategy in a distributed system, MA plays a role which beforehand identifies and classifies tasks distributed to every PE, makes a prediction by history load state of every PE, then forecasts prospective load information of every PE to dynamic allocated tasks, and at last finishes the dynamic load balancing process.

2. TASKS CATEGORY

When we decide to select which load balancing strategy, we should consider load condition of PE which it generally is confirmed by resources availability. Resources of PE have several categories: (1) cache in memory, i.e. Memory usage (2) Disk utilization (3) CPU Performance (4) I/O Performance. Generally, four factors of the resource are very important to all tasks running on PE. Speed of access memory is great high then speed of access disk and resources expended by different tasks are dissimilar, so it is propitious to take a load balancing strategy that we make a tasks category by these used resources.

We can define usage of four resources or called load of resources in PEs: L_M (Memory load), L_D (Disk Load), L_C (CPU Load), L_I (I/O Load). At the same time, we can define four coefficients: α , β , γ , μ , and they respectively are Memory Load coefficient, Disk Load coefficient, CPU Load coefficient, I/O Load coefficient. Then cost load of PEs can be computed as follows:

$$L = \alpha * L_M + \beta * L_D + \gamma * L_C + \mu * L_I \quad (1)$$

$$\alpha + \beta + \gamma + \mu = 1 \quad (2)$$

Where L is the total load of every PEs.

At the different application environment, for instance, the distributed web server or distributed computing system, the value of $\alpha, \beta, \gamma, \mu$ is not same, so we must make a category by specific application. In a general way,

* This research is sponsored by STPP (Science and Technology Planning Projects) [pc200320007] of Zhuhai city, in China.

3. LOAD-BALANCING STRATEGY

Load condition of PEs can be obtained by MA.

At first, for expressly depicting this model, we can define several terms:

Probing Interval: it is an interval that MA probe and collects load information of PE between this time and next. This interval has a direct relation to network delay of distributed system.

Prediction Window Set: a window is a set that in which all load information will be collected at N probing interval. Prediction window set size is relative to forecasting accuracy, which is to say, value of N is the more great, history load information included window set is the more many, and forecasting accuracy is the more high, but CPU load used by prediction is aggravated. The other way round, if window set size is small, forecasting accuracy will decrease.

Threshold Level: it is a parameter that which is give at the beginning and can reflect load state of PEs. It include High threshold level and Low threshold level. If load value of PEs is greater than High threshold level, PE is overload and can not accept more tasks. But if load value of PEs is less then Low threshold level, PE is underload and can take tasks of other PEs.

Rate of Change Load Balance: it is rate that load information of PEs collected by MA change between this time and next time. If rate of change load balance is positive value, we can know that load of PEs is aggravating. by contraries, if rate of change load balance is negative value, load of PEs is lighter.

At the initial condition, all of tasks will be stochastically dispatched in this distributed system, because load of all PEs is zero at the beginning. When the system begins to run, local agent of PE will work. At the every probing interval, it will collect load information of system resources, and send these load information to other PEs by MA, and then other PEs can predict futural load state of every PEs by these information. This is to say, every PEs in system will keep all load information of other PEs, so there are several table in every PEs: history load information table (HLI table), overload PE table (OLPE table), underload PE table (ULPE table). It keeps all history load information of all PEs in HLI table, and this table consists of four rows: memory load row, disk load row, CPU load row, I/O load row. As follows:

Table1.History load information table (HLI table)

	PE 1					PE 2						
	L _C	L _I	L _D	L _M	L	L _C	L _I	L _D	L _M	L	L _C	L _I	L _D
T1	30 %	50 %	20 %	45 %	40 %	44 %	25 %	67 %	54 %	50 %
T2	70 %	30 %	60 %	55 %	50 %	63 %	37 %	49 %	20 %	46 %
...
...

Where L_M is memory load value, L_D is disk load value, L_C is CPU load value, L_I is I/O load value.

It keep all overload PEs in system in OLPE table, so we can

treat this table as a overload PEs queue in which load value of PE is greater than High threshold level. It keep all underload PEs in system in ULPE table, so we can treat this table as a under PEs queue in which load value of PE is lower than Low threshold level. In tow table, the number of row is decided by task category of this system. Every row keeps respective PEs for every task. As follow:

Table2. OLPE table

T1	T2	T3	...
PE 1	PE 4	PE 2	...
PE 3	PE 6	PE 5	...
...

Table3. ULPE table

T1	T2	T3	...
PE 2	PE 7	PE 3	...
PE 4	PE 5	PE 6	...
...

Where T1, T2, T3 is all task category of this system.

Load balancing strategy can be separated to five steps:

(1) When a task request arrived to a PE, at first, local agent of PE must estimate task style and know that this task use which resource, and then search load information of relevant resources of itself PE. If load value of relevant resources is not over high threshold level, we can suppose that this task can be run in this PE. Then local agent can accept this task.

(2) If load value of relevant resources is over high threshold level, we can suppose that this PE cannot run the new task. So local agent will search the PE that can deal with this task in relevant task row of ULPE table. If local agent find a appropriate PE, a MA will be created in system and dispatch this task to another PE found in table. Then another PE will run step (1).

(3) If there is empty in relevant task row of ULPE, we can say that there is not an underload PE in system. Then local agent will select a PE from other non-overload PEs in system and dispatch this task to it. A MA will be created in system and dispatch task to another PE. Then another PE will run step (1).

(4) If there is not an appropriate PE in system, local agent will increase high threshold level and then run step (2).

(5) When load value of a PE is under low threshold level, a MA of requested task would be created. It will search all PEs in ULPE table and send a task request to other overload PE for reducing load of other PEs. If there is empty in ULPE table, we can adjust high threshold level to low.

At all steps, searching algorithm is fastest fit searching or best fit searching.

4. PREDICTION MODEL

It is prediction model that we make an analysis of time series for load information series observed and forecast future value of time series by mathematical model obtained. In this paper, we make a prediction by load information in HLI table. Prediction includes two aspects: qualitative prediction and quantitative prediction.

(1) Qualitative Prediction

In this model, we combine fuzzy mathematics with rate of change load balance. It is supposed that window set size is N . We partition load value into three phases: zero load to low threshold level (underload phase), low threshold level to high threshold level (normal phase), high threshold level to fully load (overload phase). Load value of PEs fall into which phase as follows:

$$P_{HL} = \left\{ \sum_{i=1}^N (\alpha \bullet f(L_M) + \beta \bullet f(L_D) + \gamma \bullet f(L_C) + \mu \bullet f(L_I)) \right\} / N \quad (3)$$

$$P_{LL} = \left\{ \sum_{i=1}^N (\alpha \bullet g(L_M) + \beta \bullet g(L_D) + \gamma \bullet g(L_C) + \mu \bullet g(L_I)) \right\} / N \quad (4)$$

Where P_{HL} is a probability that load value of PEs is greater than high threshold level in N sample. P_{LL} is a probability that load value of PEs is smaller than low threshold in N sample. Function $f(A)$ and function $g(A)$ are characteristic function. When A is more than a value given, function value is 1, or else, function value is 0.

At the same time, we must consider effect of rate of change load balance. If rate of change load balance is positive value and load state of PE is changed from underload into overload, we can know that PE accepts a (or some) task(s) and load of PE sharply rise. So load state of PE is overload. Contrarily, if rate of change load balance is negative value and load state of PE is changed from overload into underload, we can know that PE finish a (or some) task(s) and load of PE fall. So load state of PE is underload.

(2) Quantitative Prediction

It makes a prediction by index smooth method. Index smooth method is an approach of time-series analyses. We can regard load information as time series, y_1, y_2, \dots, y_n , then forecast load value at i time is obtained by smooth equation as follow:

$$y_{i+1} = s_i = \alpha y_i + (1 - \alpha) s_{i-1} \quad (5)$$

Where α is a weight smooth constant that its value is 0 to 1. S_i is a index smooth value at i time.

In this method, α selection decides forecasting accuracy. α value define a proportion between new forecasting data and old forecasting data. α value is greater, then proportion of new forecasting data is more, proportion of old forecasting data is little. the reverse is true. This prediction method is effective for short term prediction. So it is best fit to fast and dynamic prediction.

5. TABLE UPDATE STRATEGY

In this distributed system, there is not a global timestamp, and every PEs keeps a itself timestamp. At the time which system make initialization, timestamp of every PEs must have a initialization process. At the same time, timestamp will be automatically accumulated at every interval.

Table update strategy includes two parts: time update strategy and trigger update strategy. Each part consists of local table update and remote table update.

(1) Time Update Strategy

Local table update: at the end of every probing interval, local agent will probe and collect load information of all resource

of PEs, and then it will update local HLI table. Thereafter, local agent will predict load state of PE at next interval by history load information in window set and update corresponding OLPE table and ULPE table.

Remote table update: at the same time, local agent will dispatch a MA that in which keeps load information of all recourse and itself timestamp to other PEs. This MA will inform other PEs to update their HLI tables. When other PEs receive this message send by MA, local agent of other PEs will validate whether the timestamp kept in the message is up to date. If yes, HLI table of other PEs will be updated.

(2) Trigger Update Strategy

Local table update: when local agent finds that there is empty in OLPE table, it will reasonably reduce high threshold level, and predict load state of overall system again, and then update OLPE table and ULPE table. When local agent finds that there is empty in ULPE table, it will reasonably increase low threshold level, and predict load state of overall system again, and then update OLPE table and ULPE table.

Remote table update: when a PE state change from underload condition or normal condition to overload condition, a MA which keep overload value and timestamp will be created, and move to other PEs to inform them to update this PE state. Local agent of other PEs will validate whether the timestamp is up to date. If yes, it updates OLPE table and ULPE table. When a PE state change from overload condition or normal condition to underload condition, a MA which keep underload value and timestamp will be created, and move to other PEs to inform them to update this PE state. Local agent of other PEs will validate whether the timestamp is upto date. If yes, it updates OLPE table and ULPE table.

6. SIMULATED IMPLEMENTATION AND CONCLUSION

Base on this prediction model, we employed a platform used Aglets mobile agent (IBM Corp.) and simulated a distributed system condition in a LAN (Local Area Network). On this platform, we developed a software package DAPT (Dynamic Adaptive Prediction Toolkits) for load balancing.

Factors which affect forecasting accuracy and load balancing performance include: (1) probing interval, (2) window set size, (3) task style estimation, (4) task continuity, (5) network delay.

Using the experiment environment, we performed the simulation for FFT algorithm. The result is shown in the figure below.

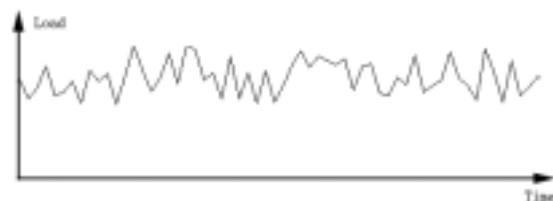


Fig. 1 The overall load on PE without prediction



Fig. 2 The predicted system load from dynamic prediction

In this experiment, Probing interval T is 10m, and ΔT is 2m-3m approximately. It's shown in the figure 1 and figure 2 that the two curves are pretty close to each other. The CPU overhead for prediction is below 1% when the system is in smooth state, while 4% or so while in the adjusting state. It's shown from the above that dynamic prediction method produces faster and accurate prediction with less consumption of system resource and adapts well, thus is a better method for load balancing. It achieve dynamic load balancing goal and level of resources utilization is heightened.

7. REFERENCES

- [1] Luis Miguel Campos *, Isaac D. Scherson " Rate of change load balancing in distributed and parallel systems" Parallel Computing 26 (2000) 1213-1230
- [2] Wei JIE Wentong CAI Stephen J. TURNER "Dynamic Load-Balancing Using Prediction in a Parallel Object-oriented System" 2001 IEEE
- [3] Li Shuangqing Gu Ping Cheng Daijie "Analysis and Research on Load Balancing Strategy in Web Cluster System" 2002.19 Computer Engineering and Applications
- [4] Rich Wolski, Neil T. Spring, Jim Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. Journal of Future Generation Computing Systems. October, 1999, Volume 15, Numbers 5-6, pp. 757-768.
- [5] Peter A. Dinda David R. O'Hallaron. An Evaluation of Linear Models for Host Load Prediction. An Evaluation of Linear Models for Host Load Prediction (1998). Nov. 1998. CMU-CS-98-148.



Yang Yongjian is a Full Professor of Computer College, the vice Dean of Software College in Jilin University. He is also a Committeeman of Directing Committee for Computer Teaching of National Education Ministry, the Director of Key Laboratory of Computer Communications of Ministry of Information Industry, and a syndic of

Computer Academy of Jilin Province. He received the master degree of Computer Application from Beijing University of Posts and Telecommunications in 1991. Since then, he has been studying Computer Network Communications. He has presided more than 10 important projects, and published one book and over 30 journal papers.

Comparison of Missing Data Estimation Methods in Satellite Information for Scientific Exploration

Zhao GuangHui, Song HuaZhu, Xia HongXia, ZHONG Luo
 School of Computer Science and Technology, Wuhan University of Technology
 Wuhan, Hubei, China, 430070
 Email: zhaogh@mail.whut.edu.cn Tel: +86 (0)27-87290674

ABSTRACT:

The study on Missing Data Estimation Methods plays an important role in satellite information for scientific exploration. In order to study the pollution of the earth atmosphere, many pictures from different angels of atmosphere have been taken by science exploring satellite. However, many data sent from satellite are lost due to the effect of satellites' orbits difference, weather conditions and photography tools. Several general methods such as K-Nearest Neighbor(KNN), Average Value Method and Single Value Decomposition(SVD) are introduced and compared in this paper. KNN is characterized by predicting accurately missing data that are classified by a small set. Experimental results show that the average value method's effect is not evident; SVD also has some limitations and KNN can predict data loss rapidly and accurately. It also suggests satellite information is strongly interrelated with space and time.

Keywords: satellite information, data loss, K Nearest Neighbor (KNN), Singular Value Decomposition (SVD)

1. INTRODUCTION

With the development of science and technology, human demands more from their living environment. At the same time, the research of environmental protection is in the ascendant. In order to study the pollution of the earth atmosphere, people have applied exploring satellite to take a lot of photos of the atmosphere and sent large quantities of data. Some data get lost and it is caused by many reasons. For example, the difference of satellites' orbits brings about data loss of adjacent latitudes and longitudes; when satellite meets different weather conditions while circling the earth, some data are often lost. Data loss is inevitable when the resolving power is low or it can't adapt to the change of atmospheric layers or there is dust on the camera lens. A lot of lost data exert negative influence on the study of satellite data, so studying and predicting these lost data have great significance.

The analysis of real-time satellite data suggests many reasons which cause data loss. Many factors function synchronously and it is impossible to make sure which is of primary importance. As a result, people often mark those lost data and take them out from original satellite data analysis in order to get new and complete data. But the operation usually loses some important information concealed in lost data. Some common methods, such as Principal Component Analysis, Singular Value Decomposition, require complete data. A solution to data loss is to conduct experiments repetitively, but the method costs much and because of incidental factors, it cannot effectively handle the problem of data loss. Currently, study in this field is at elementary stage and the frequently used methods are: Least Squares method, Iterative Analysis

Variable, Random Ratiocination and K-Nearest Neighbor method are adopted.

This paper makes comparisons between K-Nearest Neighbor(KNN), Average Value method and Singular. Value Decomposition(SVD) and proposes the corresponding results. In the end, the paper makes a conclusion that KNN is the best effective method.

2. METHODS^[1]

2.1 Average Value Method

Because the same attributes in satellite data have similar characteristics. Lost data of the corresponding attribute can be predicted according to the average value of its attribute. It is a common and easy method which is often adopted in project community. In the space of $n \times m$ swatch data X , the missing data of the i attribute is y_i . The relevant predicting formula is as follows:

$$\hat{y}_i = \frac{1}{n} \sum_{j=1}^n x_{ji} \quad \text{Eq.1}$$

In the expression, $1 \leq i \leq m$, x_{ji} means the i attribute value of the j swatch data.

2.2 The K-Nearest Neighbor Method

Because there are some relations between satellite data, the method based on KNN can combine the closely-associated data effectively and use the association to predict the missing data. Every training data sample represents a dot in an n -dimension data space. Then, the whole data set generates an n -dimension model data space.

When predicting lost data, the classifier based on KNN will search K training data samples that are closest to lost data in the model data space. And then, it predicts corresponding lost data with the average value of K samples. Euclid distance is used to measure the distance between data samples. Its expression as follows:

$$\text{dist}(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad \text{Eq.2}$$

In the expression, X means training data sample, Y is lost data, x_i and y_i respectively express the relevant entireness data attribute value of X and Y .

The essential of the method based on KNN is to search K samples that are closest to the missing data and predicts lost data with their corresponding average value. In the experiment, the weights of the nearest sample data are used. The weight of every sample data is confirmed by comparability of some attributes of missing data. Some characteristic parameters, such as Pearson relation, Euclid distance and the least variance are calculated and finally Euclid distance is regarded as a very

exact expression which is sensitive to satellite data.

2.3 The Single Value Decomposition (SVD) Method

The basic method of SVD as follows:

- (1) Learning character attribute from entire data;
- (2) Regressing the character attribute value of entire data and predicting the missing data value in the position of missing data by means of regressing method. Meanwhile the number of character attribute value must be less than that of entire data.

In the method, orthogonal model set can be obtained by using usual decomposition algorithm and the model also can express linearly the comparability of all attribute information in the data set, which shows as Eq.3. Therefore, these models are similar to primary attributes.

$$\hat{y}_R^c = U_R D_R V_R^T \quad \text{Eq.3}$$

Therein, D_R is a diagonal matrix which is made up of the order $R \leq m$ and single eigenvalue. V_R is left eigenvector of X_c and U_R is right eigenvector, they describe eigenspace with corresponding eigenvalue of diagonal matrix D_R . In order to forecast lost data, first we calculate the order R with Eq.4, making confirmed attribution of eigenvalue approach to entire data to the greatest extent. Eq.4 is as follows:

$$\min_{M \text{ rank } R} \|X^C - M\|^2 \quad \text{Eq.4}$$

Therein, X is a $n \times m$ matrix. X_c stands for entire swatch data collection, M is mean matrix. So, the order of eigenvalue matrix is rearranged based on corresponding eigenvalue. So eigenvalue matrix is identified more effectively.

Once the most important K eigenvalues in are selected, the numerical value of lost data j in satellite data i can be predicted, its operation is: regress data with the Eigenvalue K and then reconstructure j from linear combination of K eigenvalues with recursive coefficient. When recursive coefficient is determined, the numerical values of j in satellite data i and those in K Eigenvalue are not employed. SVD only deals with complete data matrix, so the original matrix X needs to be averaged to get matrix X' and then calculates the final estimated value with the maximal expectation value. Repeating the steps until the change of matrix is less than the threshold value 0.01.

3. RESULTS AND DISCUSSIONS

The above methods are all applied to restore lost data. The satellite data set is disposed with before hand to be predicted more conveniently. At first, generate a complete matrix by taking out attribute data and target data of lost data in data set and then select randomly 70% of the complete data as training data set and the rest as testing data set. All missing data form a missing data set. Then, in each method the rate of lost data is applied to renew lost data and adds them to training data set to form a new one. At last, testing data set is used to examine new training data set so as to compare rates of three methods. In addition, because the KNN and the SVD method have different parameters set, optimizing parameters is needed to make sure the accuracy of prediction.^[2]

3.1 Average Value Method

Average value method hypothesizes that the attribute of lost

data in an experiment are similar to that in other experiments. First replace attribute's lost values with 16383 and replaces object attribute's lost values with -9999. Then, replacing them with the average values of corresponding attributes or object value^[3]. The performance of the method is not good to time sequence (RMS³0.4). As we expected, average value method doesn't make full use of other attribute information, so satisfying estimated value can't be obtained^[4]. Its estimated rate of accuracy is far below KNN and SVD.

3.2 The KNN Method

It is precise to predict lost data with the method based on KNN. Based on data types and classification of lost value, the average error between estimated values and real values is about 15%. Furthermore, the method also can predict accurately the lost data, which are classified smaller. Because these smaller types are not of great contribution to Average Value method and SVD overall parameter, the two methods cannot make precise classification.

Although the accuracy of prediction can improve by using these small lost-data percentages, its performance will descend with the increase of lost data percentage. When 20% of lost data emerges, the accuracy decreases only by around 0.5%. Moreover, the method is comparatively not sensitive to exact K value in the range of 5 to 20 neighbors. If smaller neighbor number is used to make predictions, or the K value is big(>20), the performance of the algorithm would decrease. The reason is that the numerical value of neighbors is too big and is not fully related to the data predicted. So the accuracy of the expression cannot be secured. In fact, optimizing K may be selected according to the average classes of data set. When K Value increases, the effect of noises to predicted values is greater than to real values, which makes the accuracy decrease. To small quantities of data set, KNN predicts lost data precisely, too. But the method is not recommended if the attributes of the training data are less than 4.

Table1 KNN for different lost rate and rate of estimated accuracy of neighbors

neighbor number k lost rate	1%	5%	10%	15%	20%
K=1	0.7816	0.7754	0.7830	0.7810	0.7762
K=5	0.8530	0.85362	0.8527	0.8517	0.8505
K=10	0.8504	0.85081	0.8504	0.8504	0.8515
K=15	0.8466	0.84738	0.8478	0.8468	0.8461
K=20	0.8419	0.83989	0.8411	0.8400	0.8393

When the lost rate is 5% and the nearest neighbor K -is 10, most RMS data error is less than 0.25 and the predicted lost value error of KNN is as follows:

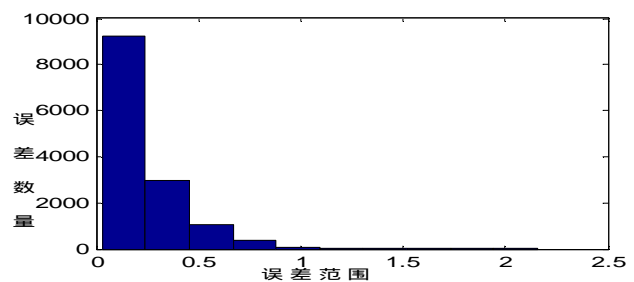


Figure1 the error distribution of KNN prediction

3.3 The Singular Value Decomposition (SVD) Method

The decisive eigenvalues 5%, 10%, 20% and 30% are used in order to confirm the optimum parameter set. When the similarity of eigenvalues is 20%, the estimation is accurate. Compared with KNN, the error curve of KNN is relatively flat between 10 and 20, and the SVD's becomes steep with the change of eigenvalues. SVD's prediction can generate higher accuracy rate than average value method and its performance is sensitive to data types to be analyzed. As for the time sequence data with noises, SVD produces satisfying results.

We replace all lost value with 0. The method is compared respectively with the above three methods. The finding of the comparison experiment is as follows:

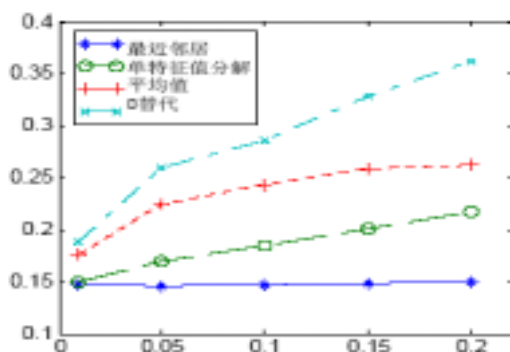


Figure2 the comparability of some method

3.4 The Complexity Analysis of three calculating methods

In matrix $n \times m$, suppose $k \ll n$. The time complexity of KNN is $O(n^2m)$ and SVD's is $O(m^2ni)$, in which i is the times that the expression works before being dealt with. Average value method's time complexity degree is $O(nm)$.

4. CONCLUSION

The KNN and the SVD provide quick and precise methods to predict satellite lost data. The two methods don't adopt traditional method in which 0 or average value is employed to replace lost values, but estimate lost values by means of the relationship of data.

The methods based on KNN and SVD has their robustness in the classification of the increased lost data. The performance of the KNN method doesn't decrease when the rate of data loss increases and it can deal effectively with types, time sequence and noise data of estimated data. It is not sensitive to the nearest neighbor parameters, but SVD doesn't handle non-optimized lost data very well. From the experiment, it can be discovered that KNN can provide accurate estimation to the small and closely related missing data in satellite data.

The KNN provides a robust, sensitive method of prediction of data loss. The method can minimize distinction between analysis methods of satellite data and provides an exact lost-value prediction method. Furthermore, prudence is needed in the selection of important geographic information and the correctness of the method needs to be constantly verified.

5. REFERENCES

- [1] Tom Mitchell, Machine Learning, McGraw Hill, 1997
- [2] Holzer-Popp, Th., Schroedter, M., Gesell, G., High Resolution Aerosol Maps Exploiting the Synergy of ATSR-2 and GOME, Earth Obs. Quarterly, 65, 19-24, 2000.
- [3] Graham Borradaile, Statistics of Earth Science Data, Springer, 2003
- [4] Jean-Yves Scanvic, Aerospatial Remote Sensing in Geology, A.A. Balkema Publishers, 1997

QoS-based Multicast Routing Optimization Algorithms for Wireless Networks*

Chen Hua, Sun Baolin

Department of Mathematics and Physics, Wuhan University of Science and Engineering
Wuhan 430073, P. R. China

Email: sun0163@163.com Tel.: +86 (0)27- 62509828

ABSTRACT

Most of the multimedia applications require strict QoS guarantee during the communication between a single source and multiple destinations. This gives rise to the need for an efficient QoS multicast routing strategy. Determination of such QoS-based optimal multicast routes basically leads to a multi-objective optimization problem, which is computationally intractable in polynomial time due to the uncertainty of resources in wireless networks. This paper describes a network model for researching the routing problem and proposes a new multicast tree selection algorithm based on genetic algorithms to simultaneously optimize multiple QoS parameters. The paper mainly presents a QoS Multicast Routing algorithms based on Genetic Algorithm (QMRGA). Simulation results demonstrate that the algorithm is capable of discovering a set of QoS-based optimal or near optimized, non-dominated multicast routes within a few iterations, even for the networks environment with uncertain parameters.

Keywords: QoS, multicast routing, wireless networks, genetic algorithm, uncertain parameters.

1. INTRODUCTION

Multicast communication is an important operation for many applications in wireless networks. For instance, soldiers wandering in a battlefield may need to keep listening to their group commanders. Similar to multicast protocols for wired networks, one of the major goals in designing multicast protocols is to reduce unnecessary packet delivery to other nodes outside the group by having only a subset of nodes participating in multicast data forwarding. Several protocols are based on constructing a tree spanning all the group members [1-5]. Flooding data packets to every node in the network can also be considered as a mesh-based protocol in that packets are forwarded over all links. To the extreme, flooding provides the most robust, but least efficient mechanism since a multicast packet will be forwarded to every node as long as the network is not partitioned, while a tree-based approach offers efficiency but is not robust enough to be used in highly dynamic environments. An efficient allocation of wireless resources to satisfy these QoS requirements is the primary goal of QoS routing. However, the interdependency and the conflicts between the individual QoS parameters make the challenge more difficult.

This paper focuses on determining multicast routes from a source to a set of destinations with strict end-to-end delay requirements and minimum bandwidth available. Though the

path determination problem with a single optimization parameter can be solved in *polynomial time*, the uncertainty of precise values of multiple objective functions make the problem a NP-hard [1-5]. The goal of this paper is to develop an algorithm to find out QoS-based multicast routes by simultaneously optimizing end-to-end delay, bandwidth provisioning for guaranteed QoS and proper bandwidth utilization without combining them into a single scalar optimization function.

The rest of the paper is organized as follows. Section 2 describes the related work. Section 3 introduces a network model. Section 4 presents the QMRGA. Analysis of convergence and some simulation results are provided in Section 5. The paper concludes and future research in Section 6.

2. THE RELATED WORK

The traditional multicast routing protocols, e.g., CBT and PIM [6-8], were designed for best-effort data traffic. They construct multicast trees primarily based on connectivity. Such trees may be unsatisfactory when QoS is considered due to the lack of resources. Several QoS multicast routing algorithms have been proposed recently. Some algorithms [2,3,5,7] provide heuristic solutions to the NP-complete constrained Steiner tree problem, which is to find the delay-constrained least-cost multicast trees. These algorithms are not practical in the Internet environment because they have excessive computation overhead, require knowledge about the global network state, and do not handle dynamic group membership. However, this algorithm requires excessive message processing overhead. Multicast routing and its QoS-driven extension are indispensable components in a QoS-centric network architecture [8-12].

To control the protocol overhead and to limit it to a tolerable level, large clamp-down timers are used to limit the rate of updates. The accuracy of network state is also affected by, for example, the scope of an update message, and the types of value advertised (exact state values or quantized values). There is a fundamental trade-off between the certainty of state information and the protocol message overhead. Moreover, in large and dynamic networks, the growth in the state information makes it practically impossible to maintain accurate knowledge about all nodes and links. Instead, the state information is usually aggregated in a certain hierarchical manner, and the aggregation process inherently decreases the information accuracy and introduces imprecision. The uncertain state information kept at each node imposes difficulty in QoS provisioning. Guerin and Orda investigated the problem of QoS routing when the state information is uncertain or inaccurate and expressed in some probabilistic manner [7,11]. Chen et al. considered a simplified probability model for link parameters [12]. They then proposed a distributed ticket-based probing routing algorithm.

*The work is supported by Key Scientific Research Project of Hubei Education Department (2003A002), NSF of Wuhan University of Science and Engineering (20032418) and Priority Discipline of WUSE (2003P1008).

In recent years, some researchers have started using evolutionary algorithms to find near-optimal solutions for different wireless networking problems, like QoS-Routing [10]. More recently, researches in determining QoS-based multicast routes clearly demonstrate the power of genetic algorithms to get a near-optimal solution satisfying the QoS requirements in computationally feasible time [9]. A little careful insight into these above optimization schemes reveals that all of them suffer from the same drawback: multiple objectives are combined to form a *scalar single-objective* function, usually through a linear combination (weighted sum) of multiple attributes. In these cases the solution not only becomes highly sensitive to the weight vector but also demands the user to have certain knowledge (e.g. priority of a particular objective, influence of a parameter over another etc.) about the problem. Moreover, in case of multi-objective optimization, a unique solution that optimizes all the objectives simultaneously will rarely, if at all, exist in practice. Conventional genetic algorithms are clearly unable to provide this flexibility to the user [10].

3. NETWORK MODEL

A network is usually represented as a weighted digraph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of communication links connecting the nodes. $|V|$ and $|E|$ denote the number of nodes and links in the network, respectively. Without loss of generality, only digraphs are considered in which there exists at most one link between a pair of ordered nodes.

Let $s \in V$ be source node of a multicast tree, and $M \subseteq \{V - \{s\}\}$ be a set of end nodes of the multicast tree. Let R be the positive weight and R^+ be the nonnegative weight. For any link $e \in E$, we can define the some QoS metrics: delay function $delay(e): E \rightarrow R$, cost function $cost(e): E \rightarrow R$, bandwidth function $bandwidth(e): E \rightarrow R$, and delay jitter function $delay-jitter(e): E \rightarrow R^+$. Similarly, for any node $n \in V$, one can also define some metrics: delay function $delay(n): V \rightarrow R$, cost function $cost(n): V \rightarrow R$, and delay jitter function $delay-jitter(n): V \rightarrow R^+$. We also use $T(s, M)$ to denote a multicast tree, which has the following relations:

- 1) $delay(p(s, t)) = \sum_{e \in P(s, t)} delay(e) + \sum_{n \in P(s, t)} delay(n)$.
- 2) $cost(T(s, M)) = \sum_{e \in T(s, M)} cost(e) + \sum_{n \in T(s, M)} cost(n)$.
- 3) $bandwidth(p(s, t)) = \min\{bandwidth(e), e \in P(s, t)\}$.
- 4) $delay-jitter(p(s, t)) = \sum_{e \in P(s, t)} delay - jitter(e) + \sum_{n \in P(s, t)} delay - jitter(n)$.

where $p(s, t)$ denotes the path from source s to end node t of $T(s, M)$.

Definition 1. QoS-based multicast routing problem deals mainly with some elements: Network $G=(V, E)$, multicast source $s \in V$, the set of end nodes $M \subseteq \{V - \{s\}\}$, $delay(\cdot) \in R$, $delay-jitter(\cdot) \in R^+$, $cost(\cdot) \in R$, and $bandwidth(\cdot) \in R$. This routing problem is to find the $T(s, M)$ which satisfies some QoS constraints:

- 1) Delay constraint: $delay(p(s, t)) \leq D$
- 2) Bandwidth constraint: $bandwidth(p(s, t)) \geq B$

- 3) Delay jitter constraint: $delay-jitter(p(s, t)) \leq J$

Meanwhile, the $cost(T(s, M))$ should be minimum. Where D is delay constraint, B is bandwidth constraint and J is delay jitter constraint. In the above QoS constraints, the bandwidth is concave metric; the delay and delay jitter are additive metrics. In these metrics, the multiplicative metric can be converted to the additive metric.

In order to develop such a multi-objective QoS-Routing algorithm, we focus our ideas on determining the multicast routes satisfying the three major objective parameters, namely: (i) end-to-end delay requirement, (ii) bandwidth provisioning for guaranteed QoS and (iii) proper bandwidth utilization.

4. QMRGA

Genetic algorithms are based on the mechanics of natural evolution. Throughout their artificial evolution, successive generations each consisting of a population of possible solutions, called individuals (or chromosomes, or vectors of genes), search for beneficial adaptations to solve the given problem. This search is carried out by applying the Darwinian principles of "reproduction and survival of the fittest" and the genetic operators of crossover and mutation which derive the new offspring population from the current population. Reproduction involves selecting, in proportion to its fitness level, an individual from the current population and allowing it to survive by copying it to the new population of individuals. The individual's fitness level is usually based on the cost function given by the problem (e.g., QoS multicast routing) under consideration. Then, crossover and mutation are carried on two randomly chosen individuals of the current population creating two new offspring individuals. Crossover involves swapping two randomly located sub-chromosomes (within the same boundaries) of the two mating chromosomes. Mutation is applied to randomly selected genes, where the values associated with such a gene is randomly changed to another value within an allowed range. The offspring population replaces the parent population, and the process is repeated for many generations. Typically, the best individual that appeared in any generation of the run (i.e. best-so-far individual) is designated as the result produced by the genetic algorithm.

QoS based multicast route discovery

The function QoS based multicast route discovery takes the source node v_s and a specific number of multicast destination nodes, say, $v_{d1}, v_{d2}, \dots, v_{dn}$ as input. It calls the function *path finding()* to find all possible multicast paths from v_s to each of $v_{d1}, v_{d2}, \dots, v_{dn}$, using the basic *depth first search* (dfs) algorithm. This gives birth to the initial set of multicast trees. The primary objective of our algorithm is to find the multicast trees from this set, which will satisfy the multiple constrained QoS parameters.

Population Initialization

Since the underlying approach is based on multi-objective genetic algorithms (MOGA), our next step is to map the problem in a search space suitable to MOGA. Each of all the generated multicast trees is mapped to a string consisting of the sequence of nodes along the path from the source v_s to each of the destinations $v_{d1}, v_{d2}, \dots, v_{dn}$. To mark the end of a path from a source to a single destination, we use -1 as *sentinel*. Figure 1 below gives a clear view of this scenario where a multicast tree is represented by a string. The set of all such strings constitute the *initial population*. The size of this population *popsizes* depends on how the strings are created,

which in turn depends on the network topology and the number of multicast destination nodes.

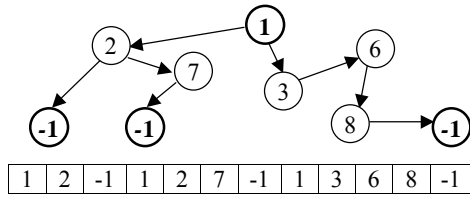


Figure 1. Representation of the Multicast tree and its encoding scheme

Fitness Sharing Function

The fitness function interprets the chromosome in terms of physical representation and evaluates its fitness based on traits of being desired in the solution. But, the fitness function must accurately measure the quality of the chromosomes in the population. The definition of the fitness function, therefore, is very critical.

Fitness function should describe the performance of the selected individuals. The individual with good performance has high fitness level, and the individual with bad performance has low fitness level. Let links be service queues where packets to be transmitted get serviced. In most cases this service can be assumed to follow Poisson distribution. The service time should follow an exponential distribution. Let the delay for link l be denoted by the variable d_l , which is a random variable following exponential distribution with parameter equal to λ . So the delay over a path consisting of k links would be the sum of k independent random variables all having the same exponential distribution and so would follow an Erlang-K distribution. From the definition of Erlang-K distribution we get that the probability that the delay (d_p) over a path P of length k is less than t is given by the following equation:

$$Pr(d_p < t) = \frac{\lambda^k t^{k-1} e^{-\lambda t}}{(k-1)!}.$$

From the classical probability theory we can say that the probability that the delay (d) of the selected multicast tree (T) will meet the specific delay constraint can be obtained by taking the product of delay over individual paths in that multicast tree:

$$Pr(d_T < t) = \prod_{p \in T} Pr(d_p < t).$$

To find an optimal path, our objective is to maximize this probability of satisfying delay requirements. The measure of the bandwidth guarantee can be obtained by assuming a similar model for the network links. If the service rate or the transmission rate, which is basically a measure of link bandwidth, is assumed to follow a poisson distribution, the probability that a link $l \in E$ can provide a bandwidth of B is given by

$$Pr_l(B) = \frac{\lambda^B e^{-\lambda}}{B!}.$$

We can now say that the probability with which the bandwidth guarantee of B is satisfied for an entire multicast tree (T) is given by:

$$Pr_T(B) = \prod_{l \in T} Pr_l(B).$$

The normal conjecture is that the path which is capable of providing with greatest residual bandwidth is the best choice. The total residual bandwidth in the network after allocating bandwidth for a multicast $T(s, M)$, is given by $\sum_{l \in E} (c_l - b_l)$, where c_l is the capacity of a link $l \in E$ and b_l is the bandwidth

allocated for all the paths in the multicast $T(s, M)$, along the link l . Obviously, b_l is 0 if $l \notin p$ where $p \in T$. The fraction of total bandwidth available as residual bandwidth is given as:

$$R(T) = \frac{\sum_{l \in M} (c_l - b_l)}{\sum_{l \in M} c_l}$$

The *fitness sharing* function of QMRGA can be defined as follows:

$$f'(x_i) = \frac{f(x_i)}{m_i}$$

To incorporate this idea of *fitness sharing* we compute the value of *niche count* for every individual string present in the population, as:

$$m_i = \sum_{j=1}^{popsize} SH[d_{s1}, d_{s2}]$$

where $d_{s1,s2}$ is the distance between individuals $s1$ and $s2$ and $SH[d_{s1,s2}]$ is sharing function. For simplicity, triangular sharing function has been used:

$$SH[d_{s1,s2}] = \begin{cases} 1 - \frac{d_{s1,s2}}{\sigma_{share}} & d \leq \sigma_{share} \\ 0 & d > \sigma_{share} \end{cases}$$

Here σ_{share} is the niche radius, and it is a good estimate of *minimal separation* expected between the goal of solutions. Individuals within σ_{share} distance of each other degrade each other's fitness, as they are in the same niche.

The phenotypic distance between two strings is nothing but the Euclidian distance between their different fitness values:

$$d_{s1,s2} = \sqrt{(\sigma_{delay_{s1,s2}})^2 + (\sigma_{bw_{s1,s2}})^2 + (\sigma_{bit_{s1,s2}})^2}$$

where $\sigma_{delay_{s1,s2}} = Pr(d_{s1} < t) - Pr(d_{s2} < t)$, $\sigma_{bw_{s1,s2}} = Pr_{s1}(B) - Pr_{s2}(B)$, $\sigma_{bit_{s1,s2}} = R(s1) - R(s2)$.

Similarly, we compute the *niche radius* σ_{share} as some fraction of the maximum separation possible in the population, i.e.

$$\sigma_{share} = \frac{\sqrt{(\sigma_{delay_{max}})^2 + (\sigma_{bw_{max}})^2 + (\sigma_{bit_{max}})^2}}{4}$$

where $\sigma_{delay_{max}} = Pr_{max}(d < t) - Pr_{min}(d < t)$, $\sigma_{bw_{max}} = Pr_{max}(B) - Pr_{min}(B)$, $\sigma_{bit_{max}} = R_{max} - R_{min}$.

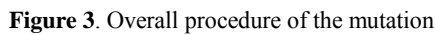
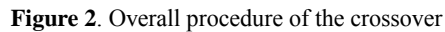
Crossover and Mutation Operations

As the algorithm executes, at every iteration we get a set of *non-dominated* strings whose fitness values represent the *Pareto-optimal* solutions for that iteration. The crossover and mutation operations are the same as normal genetic algorithms. But, it must be made sure that these operations must not produce any illegal paths. A close look into the structure of the chromosome in figure 1 reveals that these genetic operations can not be performed on any arbitrary gene (network nodes), as that can give birth to some paths which do not exist at all.

Both the crossover and mutation operations can only be performed at the end of an existing path, i.e. immediately after a particular *sentinel*, represented by -1. To give an equal probability to all such possible crossover and mutation points, we randomly select one such point. The crossover

convergence and the random selection which may tend to purity in the evolution process, and make it difficult to find the global optimal solution problem. According to the merits about championship selection, changeable length chromosomes, crossover and mutation operations, etc. in literature [13], the genetic algorithm can converge to the global optimal solution.

Simulation experiments are performed over a network of 25 nodes, consider a link from i to j that has a QoS descriptor denoted as (d, j, b, c) , where d is delay, j is delay jitter, b is bandwidth, c is link cost. In this simulation experiments, delay constraint $D=20$, delay jitter constraint $J=30$, bandwidth constraint $B=40$ and the number of multicast destination nodes being 5 $\{4, 9, 14, 19, 24\}$. We can use C program to design a genetic algorithm to simulate. Following figure 4 is the network topology structure used in the simulation example. Figure 5 shows the multicast tree found by the algorithm for the indicated form source to destination set. The multicast QoS routing protocol designed by us tries to maximize the probabilities of meeting end-to-end delay, bandwidth requirement and bandwidth utilization within a few generations by building the Pareto optimal fronts. For simplicity, we assume that QoS constraints of all leave nodes (end nodes) are the same. Figure 6 and figure 7 shows the varied curves of cost and delay of the multicast tree with the increasing genetic algebra. Figure 6 and figure 7 tells us that QMRGA can achieve global optimal solution or near optimal solution quickly.



Proof. The genetic algorithm has following merits: (1) Changeable length chromosome encoding method based on routing expression is used; (2) Crossover probability between (0,1); (3) Mutation probability between (0, 1), randomly choosing some individuals from the population with championship selection method; (4) It the individuals which has higher fitness level in the population, caused these individuals reproduce rapidly in population, easily produced

Figure 4. Network topology structure

Multicast applications involving real-time audio and/or video transmissions require strict QoS constraints (end-to-end delay bound, delay jitter and bandwidth availability) to be met by the network. To guarantee real-time delivery of multimedia

packets, a multicast channel needs to be established in advance by using a path selection policy that takes into account the QoS constraints. Among numerous advances in high-performance networking technology, the multicast routing with QoS constraints has continued to be a very important research area.

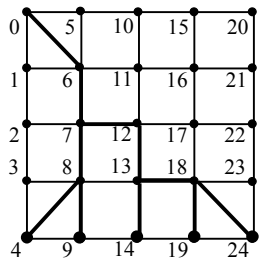


Figure 5. Genetic algorithm generate multicast tree

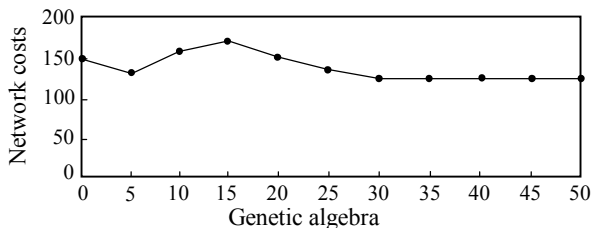


Figure 6. Network costs vs. genetic algebra

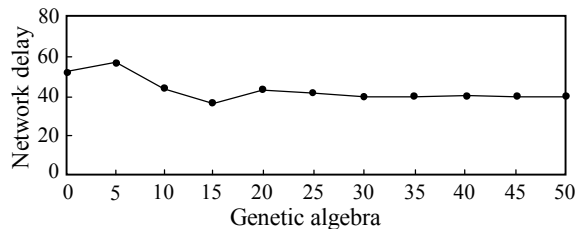


Figure 7. Network delay vs. genetic algebra

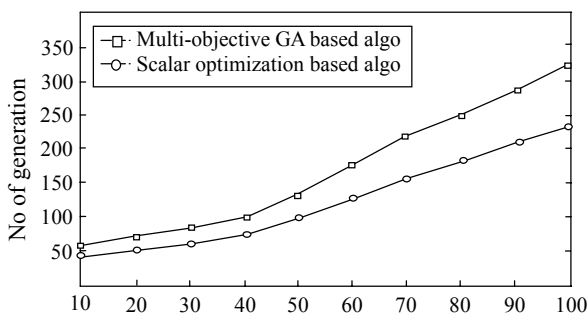


Figure 8. Performance of the algorithm with increasing number of nodes

This paper has discussed the multicast routing problem with multiple QoS constraints in the wireless networks environment with uncertain parameters. On-demand multicasting with guaranteed QoS is currently an active area of research. Seamless transmission of wireless video traffic has already become a real challenge of current and future generation wireless systems. Researches in the QoS routings are mostly done to optimize these QoS parameters by combining their different, conflicting characteristics into a single scalar function with the real intuition and logic behind the combinations being often fuzzy. The QMRGA can both optimize the network resources such as bandwidth and delay

and converge to the optimal or near-optimal solution within few iteration, even for the networks environment with uncertain parameters. The incremental rate of computational cost can close to polynomial and is less than exponential rate. Simulation results delineate the efficiency, performance and scalability of the protocol. Our future interest is to mathematically model this protocol to analyze its performance and complexity. Finally, we think our work will be helpful in solving some new problems in the domain of QoS Routing.

In a word, the deep research of QoS constraint multicast routing will increase the technology of high performance network routing system, and it will be widely applied in video, multimedia broadcasting and distance education fields, etc.

7. REFERENCES

- [1] Li Layuan and Li Chunlin, "The QoS routing algorithm for ATM networks", *Computer Communications*, Vol. 24, No.3-4, 2001, pp. 416-421.
- [2] Li Layuan and Li Chunlin, "The QoS-based routing algorithms for high-speed networks", *Proc of WCC*, Aug. 2000, pp. 1623-1628.
- [3] Li Layuan and Li Chunlin, "A multicast routing protocol with multiple QoS constraints", *Proc of WCC*, Aug. 2002.
- [4] M. Liu, R. R. Talpade, and A. McAuley, "AMRoute: Adhoc Multicast Routing Protocol", Tech. Rep. 99, The Institute for Systems Research, University of Maryland, 1999.
- [5] C. Wu and Y. Tay, "AMRIS: A Multicast Protocol for Ad hoc Wireless Networks", in *IEEE Military Communications Conference (MILCOM)*, (Atlantic City, NJ), November 1999, pp. 25-29.
- [6] Bin Wang and Jennifer C. Hou, "Multicast routing and its QoS extension: Problems, algorithms, and protocols", *IEEE Network*, Jan/Feb, 2000, pp. 22-36.
- [7] Roch A. Guerin and Ariel Orda. "QoS routing in networks with inaccurate information: Theory and algorithms", *IEEE/ACM. Trans. On Networking*, Vol. 7, No.3, June, 1999, pp. 350-363.
- [8] Moses Charikar, Joseph Naor and Baruch Schieber, "Resource optimization in QoS multicast routing of real-time multimedia", *Proc of IEEE INFOCOM*, 2000, pp. 1518-1527.
- [9] N. Banerjee and S. K. Das, "Fast Determination of QoS-based Multicast Routes in Wireless Networks using Genetic Algorithms", *International Conference for Communication, ICC-2001*, 2001.
- [10] F. Xiang, L. Junzhou, W. Jieyi and G. Guanqun. "QoS routing based on genetic algorithm", *Computer Communications*, Vol.22, 1999, pp. 1392-1399.
- [11] Dean H. Lorenz and Ariel Orda. "QoS routing in networks with uncertain parameters", *IEEE/ACM Transactions on Networking*, Vol.6, No.6, DEC.1998, pp. 768-778.
- [12] S. Chen and K. Nahrstedt, "Distributed QoS routing in ad-hoc networks", *IEEE JSAC*, special issue on ad-hoc networks, Aug. 1999.
- [13] Wang Xiaoping and Cao Liming. *Genetic Algorithm—Theory, Application and Software Realization*. Xi'an: Xi'an Jiaotong University Press, 2002(in Chinese)



Chen Hua received the B.S. degree in mathematics from Central China Normal University, Wuhan, China, in 1999 and M.S. candidate in computer science of Huazhong University of Science and Technology. She is currently a Lecturer at the Department of Mathematics and Physics at the Wuhan University of Science and Engineering. Her research interests include Genetic Algorithms, distributed computing.

Sun Baolin is currently an Associate Professor at the Department of Mathematics and Physics at the Wuhan University of Science and Engineering with Ph. D. candidate in computer science of Wuhan University of Technology. He has published one book and over 20 journal papers. His research interests include QoS multicast routing, Genetic Algorithms, distributed computing and protocol engineering.

A Core-Stateless Dynamic Bandwidth Allocation Mechanism Based on Resource Reservation

Liu Quan, Liang Xiaoyu, Li Fangmin

School of Information Engineering, Wuhan University of Technology
Wuhan, Hubei 430070, China

E-mail: xiaoyu_liang@sohu.com Tel: +86(0)27-87299825

ABSTRACT

This paper presents a core-stateless dynamic bandwidth allocation mechanism based on resource reservation. A stateless recursive monitoring mechanism is introduced to adjust the reservation bandwidth dynamically, which enhances the scalability and robustness of QOS. To achieve bandwidth allocation dynamically, three key techniques are developed. The first one is the lightweight certificate on control planes. The second one is stateless recursive monitoring mechanism on data planes and the last one is that the traffics are divided into marked flows and non-marked flows. Finally, the simulation results are presented and the mechanism is testified.

Keywords: resource reservation, recursive monitoring, lightweight certificates, dynamical bandwidth allocation, core-stateless

1. INTRODUCTION

The Internet has traditionally supported the best-effort service model. However, as the Internet applications on internet include not only data application in recent years, but also multimedia applications including video conference, long-range cooperating, virtual application, IP telephone etc. Especially on the intellectual control field, multimedia is becoming more extensive than before, with the advent of real-time and mission-critical Internet applications, there is an increasing need for network service providers to export high bandwidth and low delay guarantees to such services. It has become more and more important for routers providing congestion control and bandwidth allocation mechanism.

During the last decade a plethora of solutions have been developed to provide better services than best-effort. IETF has provided two network architectures for Internet QOS: One is Intserv (Integrated services) and another is Diffserv (Differentiated Services) based on Per Hop Behavior. Intserv provides QOS guarantee by bandwidth and delay and maintain per-flow state of reservation. The singlings of the control plane are finished based on per-flow state. On data plane, Intserv assumes a network architecture in which every router maintains per-flow state every packet's state. So much state maintenance information will lead to the poor performance of the network. Although Intserv can provide QOS from end to end, every router through the path must support RSVP, which will cause scalability limitations. Diffserv differs edge routers with core routers. It pushes access control to network border and core router processes each packet just according to DSCP (Differentiated Services Code Point). The above-mentioned measures in Diffserv get over the scalability limitations, but it can't provide absolute service guarantee. For instance its performance is relatively bad in assured service and premium service.

This paper presents a core-stateless dynamic bandwidth allocation mechanism based on resource reservation, which combines the characters of Intserv and Diffserv. In this mechanism, the RSVP of Intserv and the architecture of edge routers and core routers in Diffserv are reserved. A stateless recursive monitoring mechanism is introduced to adjust the reservation bandwidth dynamically, which enhances the scalability and robustness of QOS.

2. NETWORK MODEL

In this paper, three key techniques are developed. The first one is the lightweight certification in conjunction with a soft state [1] approach based control planes by not requiring routers to maintain information about every individual flow. The second one is stateless recursive monitoring mechanism on data planes by requiring routers to maintain the state of an aggregate flow, which can detect whether flows exceed their reservation. The last one is that the traffics are divided into marked flows and non-marked flows according to paper [2].

2.1 CSPAFA mechanism

The core idea of CSPAFA is SCORE (Stateless Core) [3]. It classes all flows into two kinds: marked flows and non-marked flows. The core routers divide the outgoing link bandwidth C into C_R (aggregate outgoing bandwidth of marked flows) and C_B (aggregate outgoing bandwidth of non-marked flows). The edge routers, which encode state information about flow into the packet header, process traffics regarding to per flow state. The core routers perform packet scheduling and other actions according to state information.

2.1.1 Edge router operations

Edge routers group the flows into marked flows and non-marked flows. The aggregate arrival rate r_i of the two kinds of flows is estimated by CSFQ [3] and a packet label is inserted into IP header by edge routers. At the same time, some reserved bandwidth R_i is reserved according to the service specifications. For unmarked flows, CSPAFA only inserts r_i into packet header. However, for marked flows, it inserts the proportion (R_i/r_i) and the flows share the bandwidth in proportion to R_i/r_i . The estimated rate is estimated by exponential averaging shown as formula (1):

$$r_i^{new} = (1 - e^{-T_i^k/K}) \frac{l_i^k}{T_i^K} + e^{-T_i^k/K} r_i^{old} \quad (1)$$

Where $r_i(t)$ represents the estimated rate of flow i at time t and r_i will be updated based on formula(1) and inserted into the packet header upon the reception of every packet. l_i^k and t_i^k represents the k -th sample of the interarrival time and its length

of flow i . K is a constant and $T_i^k = t_i^k - t_i^{k-1}$.

2.1.2 Core router operations

(1) Marked flows operations

Each packet of marked flow contains the ration of the reserved rate R_i and the estimated rate r_i the reserved rate R_i (R_i/r_i), where induces is to allocate bandwidth to marked flows proportionally according to the reserved rate in service specifications. The will be grater than 1 if the transmitted rate is less than the reserved rate of one flow, thus the proportion of packet lost rate is decreased accordingly. The aggregate rate is estimated also by exponential averaging, computed as formula (2):

$$R_a^{new} = (1 - e^{-T/K_a}) \frac{1}{T} + e^{-T/K_a} R_a^{old} \quad (2)$$

(2) Non-marked flow operations

The non-marked flow operations adopts fair bandwidth allocation algorithm of CSFQ. Here two parameters should be estimated: one is the arrival rate $r_i(t)$ of estimated flow and another is fair shared speed $\beta(t)$. The former is estimated by formula (1) in edge routers, and is inserted into packet header. The latter is computed as [2].

If the aggregate rate of all non-marked flows is $R_b(t)$ at time t and its estimated fair shared speed is $\beta(t)$, the outgoing rate of per-flow through outgoing link equals to $\min(r_i(t), \beta(t))$ and the aggregate rate F of all flows is function of $\beta(t)$, shown as:

$$F(\beta(t)) = \sum_{i=1}^n \min(r_i(t), \beta(t)) \quad (3)$$

If $R_b(t) \leq C_B$, defined $\beta(t) = \max(r_i(t))$, no packet will be lost. If $R_b(t) > C_B$, choose $\beta(t)$ as the exclusive evaluation of $F = C_B$. For $r_i(t)$ has been already insert into the packet header, $\beta(t)$ can be obtained. This way needs to track the arrival rate $r_i(t)$ of every flow in core routers. R_b and F are estimated by formula (2), without taking the lost packets into account when calculating F .

2.2 Control Plane

The main function of the control plane is to perform admission control and establish and maintenance the state needed by data plane. There are routing protocols and singling protocol existed in control plane. The present main routing protocols include BGP (Border Gateway Protocol) used among autonomous systems, RIP(Routing Information Protocol) used inside in autonomous system and OSPF(Open Shortest Path First). The singling protocol includes accession control and route pinning. ATM UNI and RSVP involved in the former and the latter deals with how to ensure all packets of per flow transmit through the way determined by accession control in order to assure QOS.

When adopted above-mentioned CSFPA mechanisms, the actual bandwidth we can get is $(R_i/R_a) * C_R$ if the estimated rate r_i exceeds the reserved rate R_i . Then the excess part $r_i - (R_i/R_a) * C_R$ will be lost. In the following, a measure got over the above-mentioned shortcoming will be introduced—recursive monitoring mechanism

2.2.1 Admission Control

To perform bandwidth admission control, each router maintains

the total amount of bandwidth reserved so far on an outgoing link, C_R . Upon receiving a reservation request for bandwidth r , the router simply checks whether $C_R + r \leq C$, where C is the link capacity. If this is true, the router accepts the reservation and updates C to be $C + r$.

The challenge is to maintain an accurate value of C_R in the face of partial admission failures, message losses and flow termination, without maintaining per-flow state. To address this challenge we use a soft-state approach similar to the one proposed in [1]. Each source that is granted a reservation is required to send periodic refresh messages as long as it intends to maintain the reservation. The interval between two successive refresh messages, denoted T_{ref} , is fixed and known by all routers that implement our solution. Each refresh message contains the bandwidth reserved by the flow. The router can then compute the aggregate reservation on the output link by simply adding the values carried in all the refresh messages received during a period T_{ref} . The scheme is robust to losses and partial reservation failures because inaccuracies do not build up over time.

A potential problem with the above solution is that consecutive refreshes of a flow can arrive more than a period T_{ref} apart due to delay jitter, which will cause the wrong estimated reservation. One way to alleviate this problem is to compute the aggregate reservation over a larger period of time which is denote by T_{router} . For simplicity, choose T_{router} to be a multiple of T_{ref} , that is, $T_{router} = n_{ref} \times T_{ref}$. We also assume that the maximum jitter does not exceed T_{ref} , so that at most 1 refresh is “missed”. Further, when a router receives a reservation request for bandwidth r in the middle of a router period, (at time d from the beginning of the router period), then the router increments its total reservation estimate by $r \times [d / T_{ref}]$ not just by r .

2.2.2 Control Plane Misbehavior

The reservation estimation algorithm presented above assumes that all sources obey the control plane protocol. But the fact is not so. In this paper, three possible situations in which a source can misbehave on the control plane are discussed: (1) Sending refresh messages without having been admitted. (2) Stop sending periodic refreshes and resume sending them later without undergoing admission control again.

Scenarios (1) and (2) can be solved by lightweight certificates generated by routers in order to allow the routers to verify that a refresh message corresponds to a flow which was admitted earlier.

The key idea of lightweight certificates is as follows: Each router along the path computes and attaches a certificate to the admission request message if the admission succeeds. The request message collects the certificates from all routers along the path, and the destination sends them back to the source in a response message. Subsequently, the source sends refresh messages containing all the certificates that it had received, in the same order that they were received.

The certificate issued by a router is a one-way hash of the requested reservation amount, the flow identifier, and a router specific key. Each router on the path (which implements our solution) needs to know only its own key value, and how to access the certificate issued by it (earlier) from the control packet payload. The control packet format is as Fig1. The fields

in which these certificates are stored, in the refresh packet, is denoted C-Fields.

Whenever a router receives a refresh message, the router recomputes a certificate C_1 using the packet fields and its key. It then retrieves the certificate C_2 , which was issued by it during admission control, from the appropriate C-Field in the refresh packet. If C_1 and C_2 are identical, the router is able to conclude that (a) the refresh indeed corresponds to an earlier admitted flow, and (b) the reservation amount specified is the same as requested during admission time. In other words the certificate is "valid". Therefore the router accepts the refresh i.e., uses the specified reservation amount to update C_R .

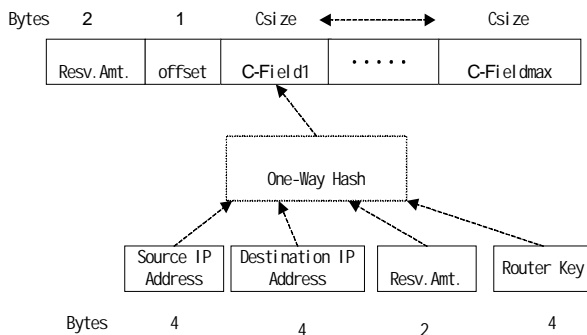


Fig. 1 Control Packet Format

As an optimization, the refresh message also contains an *Offset* field that indicates how many routers (which implement our solution) the message has traversed. This field is incremented every time the message traverses a router, and can be used by routers to efficiently access their certificate. This solution addresses scenario (1).

However, a source that was granted a reservation and stopped sending refreshes and data can use the certificates at a later time without getting readmitted. To address this problem, routers change their keys, and therefore the certificates for each flow, during every refresh period (T_{ref}). The new certificates replace the certificates in the arriving refresh packet. Sources are expected to use the new certificates in the next period. Thus, if a source stops sending refreshes it will stop receiving the new certificates. If it starts sending refreshes later, they will not contain valid certificates.

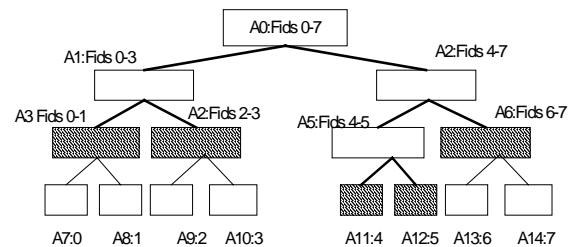
3. DATA PLANE

The goal of the data plane is to deal with data packets. The traditional routers mainly include three functions: route research, buffer management and packet scheduling. Bandwidth and delay need be guaranteed for better QoS. In this paper, data plane is to ensure that each flow receives its reserved bandwidth in the presence of misbehaving flows that exceed their reservations, without maintaining per-flow state. To address this challenge, we employ a stateless monitoring mechanism R-Mon [4] for detecting misbehaving flows.

The basic idea of R-Mon is to randomly divide the total set of flows into large aggregates and monitor the aggregates. When an aggregate misbehaves, it is recursively split into smaller aggregates that are monitored similarly. The recursion terminates when the monitored aggregate consists of a single flow.

In R-Mon algorithm, the data structures of aggregated flow use tree structure, called Aggregate-flow tree A-Tree. If the traffic of an A-flow exceeds the total reservation allocated to all flows in the A-Tree, then at least one of these flows is misbehaving.

The organization of A-Tree is shown in Fig 2. The root represents the aggregated flow consisting of all flows. The leaves of this tree represent aggregated flow each containing a single flow traversing the link. Each internal node represents an aggregated flow, which also has a set of "children" aggregated flows. So the aggregated flow denoted A-flow is a set of aggregated flow consisting of all flows. The father A-flow has two children A-flows. The number of nodes of this complete tree that can be maintained at any given time is constrained by the amount of state the router can maintain. Let B be the maximum number of A-flows that can be simultaneously monitored by the router. The sub-tree, which has as its leaves the A-flows that are currently being monitored, is called the A-tree.



(The shaded boxes represent monitored A-flows. The dark lines represent the A-tree.)

Fig. 2 Illustration of an A-Tree

The central idea of R-Mon is to recursively descend down those branches of the A-tree that lead to misbehaving flows until there is a flow. The pseudo-code for this is shown in Fig.5. To implement this router maintains two data structures:

$_monTbl$, which maintains the set of A-flows and flows that are currently being monitored. The size of this set is bounded by B .

$_alertList$, which is a priority queue that maintains the set of A-flows that are misbehaving: given two A-flows at different levels, first expand the A-flow which has a higher depth in the A-Tree; given two A-flows at the same level expand the one that misbehaves by a larger amount.

In the absence of misbehaving flows, the algorithm tries to evenly extend the complete A-Tree as deep as allowed by the bound B , i.e., up to a depth $\log_k B$, where k is the degree of the A-Tree. When one of the A-flows misbehaves, its children are added to $_monTbl$. In order to add these children, remove all A-flows that have been monitored and have not been observed to be misbehaving. If space constraints remain, need to remove A-flows (of lower priority) from $_monTbl$.

Since the removal of A-flows could result in some flows not belonging to any monitored A-flow, in practice, always monitor every A-flow at depth one (e.g., A1 and A2 in Fig.2).

To process the wrong flows by the refresh messages between the control and data planes. The data plane uses the period T_{ref} to count the number of refresh messages received by each monitored A-flow and determines the aggregate reservation of that A-flow. At the end of each such period, the router does the following for each A-flow in $_monTbl$: compare the total traffic

sent in that period by that A-flow with its aggregate reservation (bandwidth); if the A-flow is found to be misbehaving, then either its children are added to monTbl, or if the A-flow consists of a single flow, that flow is downgraded or contained. If the A-flow is just consisted of one flow, the flows whose bandwidth exceeds the reservation are changed to be non-marked flows.

4. ANALYSIS OF SIMULATION RESULTS

In ns2.26 [5], simulations are designed to evaluate our solution. These simulations aim to test the degree influenced by misbehaving flows on well-behaved flows, that is the average lost rate and the bandwidth allocation all the flows can get when there are flows exceed their reservation. The experiment model is shown in Fig 3.

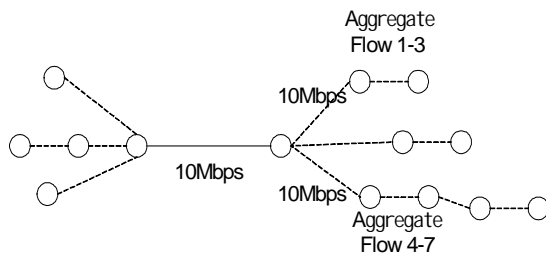


Fig.3 experiment model

The value of T_{ref} and T_{router} is 5 seconds. Data and control packets are of size 1500 and 500 bytes respectively. There are 3 flows in the marked aggregated flows and 4 flows in the non-marked aggregated flows. The bandwidths of the links are 10 Mbps and the transmission delay is 1 ms. All the flows share the 10Mbps bandwidth. Originally, 8M is allocated to marked flows and 2M is allocated to best-effort flows. The reservation of 3 marked flows is (5, 2, 1). Among the three experiments, all the marked flows send data with the estimated rate and the 4 non-marked flows are CBR flows. The results are shown in table 1, where A_i is the marked flow number of the aggregation and is the non-marked flow number of the aggregation.

Table1: the simulation results

Aggre- gation No.	Exp1 :	Exp2 :	Exp 3 :	Avg. Drop Rate	
	(5,2,1) Mbps	(3,2,1) Mbps	(5,3,2) Mbps	R-Mon	Random Sampling
A ₁	4 . 99600	2 . 99100	4 . 00100	1.6%	1.79%
A ₂	2 . 00100	2 . 00200	2 . 39800		
A ₃	1 . 00200	1 . 00100	1 . 59900		
B ₄	0 . 50000	0 . 99680	0.39840	Droptail Drop Rate	
			0.40080	23%	
B ₅	0 . 50000	1 . 00800	0.40000		
B ₆	0 . 50000	1 . 00800	0.40000		
B ₇	0 . 49920	1 . 00800	0.40000		

From the above results, the marked flows get the corresponding reserved bandwidth for the total aggregated rate doesn't exceed the reservation C_R in experiment1. In experiment2, because the total aggregated rate of the 3 marked flows is smaller than 8 Mbps, the 4 best-effort flows share the residual 2Mbps bandwidth and get 4Mbps bandwidth totally. In experiment3, the total aggregated rate 3 marked flows is 10Mbps, exceeding the reservation 8Mbps bandwidth, and there is wrong flow A_3 judged by R-Mon algorithm. The exceeding (2-1)Mbps of A_3 is changed to be non-marked flow, so there are equivalent 5 non-marked flows., whose corresponding rates are 0.39840, 0.40080, 0.40000, 0.40000, 0.40000(Mbps) respectively. That is the non-marked part of the marked flow1 gets 0.39840Mbps bandwidth and it shares the 2Mbps bandwidth with other best-effort flows fairly. At the same time, the 3 marked flows are allocated 8Mbps proportionally. Also, the drop rate of R-Mon is smaller than the random sampling algorithm.

5. CONCLUSION

A core-stateless dynamic bandwidth allocation mechanism is presented based on resource reservation, which combines the characters of Intserv and Diffserv. In this mechanism, the RSVP of Intserv and the architecture of edge routers and core routers in Diffserv are reserved. The traffics are divided into marked flows and non-marked flows, which perform different bandwidth allocation policies. The lightweight certificates and stateless recursive monitoring mechanism are adopted. These measures contribute to the reliability and robustness. The simulation results are presented by ns2.26.

6. REFERENCES

- [1] I. Stoica and H. Zhang, "Providing Guaranteed Services Without Per Flow Management", ACM SIGCOMM'99, Sep 1999.
- [2] LI Fang-Min and LI Ren-Fa, A Core-Stateless Proportional Adaptive Fair Bandwidth Allocation Mechanism Journal of Computer Research and Development , 2002 , 39 (3):269-274.
- [3] Stoica I, Shenker S and Zhang H. Core-stateless fair queuing: achieving approximately fair bandwidth allocations in high speed networks. Carnegie Mellon University, Technical Report CMU-CS-98-136, June 1998.
- [4] S. Machiraju, M. Seshadri and I. Stoica, "A Scalable and Robust Solution for Bandwidth Allocation", Technical Report UCB//CSD02 -1176, University of California at Berkeley, 2002.
- [5] "Network Simulator 2", <http://www.isi.edu/nsnam/ns/>

Liu Quan, female, professor, tutor of doctor, is the dean of the school of information engineering, Wuhan University of Technology, her research interests are information security, signal processing, communication technology, grid computing and network security.

Liang Xiaoyu, female, is a doctor candidate of communication and information system, Wuhan University of Technology. The study field is on embedded system and network security.

Customer Relation Management System Based on Mobile Internet

Di Guoqiang

Information management College, Jiangxi University of Finance & Economics

Nanchang, Jiangxi, 330013, China

Email: DyLydgq@21cn.com Tel.: 13007200302

ABSTRACT

Communicating, interacting and providing service with customer in time are basic demand of CRM for an enterprise. The cellular phone is the most popular portable intelligent communication terminal. Internet and mobile communication is the important sign of the information society, the combination of which offered convenient service way of CRM and meet the information service demand in any time and any place. The advantage of voice information service based on CTI can be reserved by Cellular phone. A small or simple service platform of CRM can be constructed alone by SMS or WAP. So enterprise should integrative or selectively utilize all kinds of technology to implement CRM according to the characteristic of that. A CRM case based on SMS is introduced finally.

Keywords: mobile Internet, CRM, CTI, SMS, WAP

1. CUSTOMER RELATION MANAGEMENT

The modern market competition is fiercer and fiercer. The competition between enterprises spreads from the products to serving, so that customer relation management is gotten extensive attention. A lot of firms are implementing or about to implement the CRM system. CRM originated at the beginning of 1980s, was the result of management idea combined with technology. It emphasize that the customer should be regarded as the center, different tactics should be taken to different customers to offer the personalized service, to improve customer's loyalty, to strengthen the competitiveness of enterprise.

The final goal of enterprise CRM system is the largest profit and amount of customer. This system need to unite the customer's information distributed in each department, to identify the customer uniquely and to provide timely service with them, so as to improve customer satisfaction, to pull up and to increase the customer, to raise the sales and profits of enterprise.

CRM is a new type of commercial mode that regarding customer as center. It is also the management way of aiming at improving the relation between the enterprise and customer. To pull up the customer means that to allow the customer communicating with the enterprise by some way that them like to, to make people getting goods info or better service from the enterprise conveniently, so as to promote the customer satisfaction degree, to increase more new users and keep the old customer. So it is an important job for enterprise. [2].

The common tools like fixed telephone or fax was used in Early CRM to communicate with customer, and then the call center based on those tools was developed. With the development and application of Internet technology, the

service based on Internet such as Email, WebPages has been introduced in CRM. Mobile communication technology is developing rapidly, is combined with Internet as mobile Internet.

2. MOBILE INTERNET

The cellular phone has been the most popular mobile communication tool nowadays. By the end of November of 2002, the propagation rate of the cellular phone in china was up to 14.95 percent according to statistics [3]. Compared with ordinary telephone, the cellular phone has all functions of the ordinary fixed telephone. It is not only a conversation tool, but also is used as the portable intelligent communication terminal because of its some peculiar function such as short message service or multimedia message service, wireless application protocol. The outstanding advantage of which is the info service whenever and wherever.

The demand of user information service with cell-phone is increasing sharply. Typically, the short message service combined with Internet, create the limitless market for information industry, and make that .COM corporation emergence in the Nasdaq stock market. Except the SMS, more function of mobile communication waits to develop.

With the development of Internet and mobile communication technology, with the popularization of the cell-phone, to offer more convenient service for customer in CRM is approved by a lot of enterprises and customer more and more.

3. MOBILE SERVICE DEMAND FOR CRM

One enterprise regarding the customer relation as an important resource will not stand its fine service ability to decline because of the objective conditions such as official condition, communication or data analysis limitation, and long time responding customer. So enterprises hope certainly to control the customer resources to respond or to meet the customer extensive demand in real time and cross over space. This is the reason to use the mobile Internet in CRM.

Common fixed phone not only can be used continue as the customer's voice terminal device in CRM system based on mobile Internet. All mobile devices such as cell-phone, mobile module, PDA, notebook computer also can be used as the terminal. But the most convenient one is the cell-phone. Not only because it is the most popularized, but also because it is an intelligent terminal of GSM. The achievement and experience based on the fixed phone information service can be used as the foundation of the CRM based on mobile Internet, some new service way and content has been increased.

The basic conversation function of the cell phone was used in CTI system set up before for customer information service. CTI refers to computer telecommunication integration developing from the traditional computer telephone integration. This technology has crossed over the computer voice treatment technology and telecommunication. With the development of computer and telecommunication network, a large number of applications on CTI technology were used in all fields, especially used in customer information service. For example call center, e-commerce, voice web site brows and so on. Because there is no difference in the mobile application based on CTI compared with regular phone, the cell-phone customer service mode based on CTI is regarded as the basic information service way in CRM. The mobile phone user can finish the information advisory service, refer questions, record voice by automatic voice guiding them action through the CTI or IVR server in CRM customer service center.

Modern mobile phone has ability of data communication with multimedia, SMS, WAP, bluetooth, so that it can make the CRM system transiting smoothly from the simple voice service mode to the mobile Internet platform.

It is an important service of mobile Internet that the short message service of GSM networks. It has developed to a new era that the multimedia message service is run from text short message service early. It can ensure information to be received because it is transmitted with the mode of store and forward. It does not need high expectations for GSM network signal, with short time of channel, low cost, high quality, roaming automatically, interacting, and receiving free no matter how long distance it is. SMS can carry many kinds of data, such as block mode, PDU mode and text mode. The content of SMS that PDU coded may be text, voice or vision. Additionally, It can be realized two different working ways, "push" and "pull".

The main short of SMS is the limit of message length; it is not more than 160 characters for each one at present. The communication with limited character not only can be realized, it also be realized that the interacting information between CRM and the users, for example the SMS game.

WAP (Wireless Application Protocol) is a promptly and safely way to access Internet or Intranet by mobile phone, similar to ordinary Internet, users can browse, query, and interchange info real-time with it. WAP bases on real-time connection. The cost of accessing Internet is high because of low transmission rate between cell phone and the GSM network at present. The mobile user can only access the WebPages based on WAP, the ordinary WebPages based on HTTP cannot be accessed because of the form difference between WAP and HTTP. The WAP is not applied more widely than the SMS because of the above limitations. But with the technology progress and performance improvement of the GSM network, it will make great progress and mobile devices will be improved. For example, 3G appear and more cell-phone supporting browser will be developed. All of these will enhance the convenience and quality of customer information service.

The physic size of cellular phone is limited though it has a lot of advantages, so that the input and output are restricted with less keys and narrow screen. Additionally, the producers produce too much more different kinds of cell

phone so that the software development supporting all kinds of cell phone is more difficult. With the price of wireless device descending and many service providers continue joining, mobile customer info service will hold more and more market.

From the analysis above, different methods of mobile service have characteristics of itself. We must effectively utilize the advantages of each way to develop CRM system.

4. THE CONFIGURATION

To service the customer with information in any time and any place, it must be combined that the mobile data communication technology with information resources and many other kinds of technology, such as Internet, multimedia technology, etc. The main important is that the customer's information resources in enterprise should be made full use of. Generally, an overall CRM system based on mobile Internet includes devices as follows:

- 1) The wide band network interconnecting equipment between the CRM customer's service center and the operator of mobile telecommunication to realize the information transmission or interaction between cellular phone and Internet, the platform of customer's information service.
- 2) Interactive voice response system (IVR) based on CTI, it can automatic respond with voice and dial customer's number including regular phone and cell-phone in batches to inform the customer of relevant service content. The user can input data by the keyboard of the cell-phone to reply or take a voice record for the system, that can allow the user authorized to access database or inquiry product information, guiding them to complete the ego service with voice or other ways at the same time.
- 3) Short message service gateway, linked with the short message service center of the mobile communication operator, interchange data with two-way and service customer with info including short message announcement, information order or issue.
- 4) WAP server offers service of mobile Internet's homepage accessing. Customers can browse the web based on WAP by cell phone.
- 5) Agent, their equipments include the telephone (digital or analog), headphone, transmitter and personal computer or terminal running the CTI application. The agent can relaxedly complete those work such as answer a call, hanging up, transferring the call and dialing out by mouse or keyboard. They can deal with the call more flexible, offer more kind and more all-sided service than the IVR.

A generally CRM system is shown as figure 1 below. While using concretely, the above-mentioned equipment can be more or less. For example, the simplest one case is the SMS to be constructed CRM platform, as the following picture shows (Fig 2). In this case, the mobile device of the host computer side may be the ordinary cell-phone or mobile module. It is a kind of CRM system with simple construct, easy to set up, low investment and costs, especially suitable for small or medium enterprises.



Figure 1: A generally CRM system

5. A CASE OF CRM SYSTEM BASED ON SMS

Yongxing Branch of Nanchang Commercial Bank is a branch bank that has the sub branch of 6 agencies, the CRM system of this branch was established in order to offer better service for the customer. After analyzing and comparing the own situation and some solutions, the way based on the SMS was chosen finally in order to stint on expenses. The system composition is as Fig. 2 shows. Mobile devices of the end in the bank adopt Nokia 6110, an ordinary cell-phone, which connected the COM interface of the computer operating the CRM software.

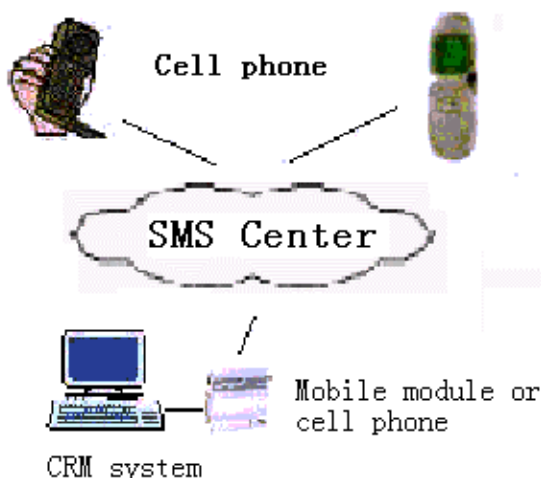


Figure 2: A simple mobile CRM service platform

Though the SMS used to construct the CRM is an extremely ordinary function of cell phone, the two-way information exchange between the bank and customers can be run, and it is not high to operate the expenses. This system offers the following service to the customer now:

- 1) The newest business info automatic notice, the bank can notify the customer automatic in time with latest business news, the batch sending is an outstanding advantage of the SMS for them to issue those message.
- 2) Account automatic notice when changed, customized by the customer, it is convenient for customer to understand his fund change and the sum in time.
- 3) Make an appointment, order the call-out or withdraw

large volume by booking the bank.

- 4) Alarm, according to the customer's requirement, it alarms them automatic by short message.
- 5) Other short message service.

6. CONCLUSIONS

The CRM system based on the advanced mobile Internet, has been greatly improved than the old one, and has caused the deep change on service content and way.

With the progress of the society, the aggravation of the competition, offering service of the whole life cycle of products for customer becomes an important respect of enterprise. The cellular phone makes it true that the dream of remote information service conveniently whenever and wherever.

The mobile information service system is the organized result of modern business administration and modern technology such as mobile communication, computer telecommunication integration and computer network. It is a front achievement of modern information technology, with the development and application of it, the customer information service will not be restricted in time or space and popularize. Not only it is the new service means for enterprise to eliminate the blind area of information interchange, but also it is significant to improve the service quality, to promote enterprise brand, to optimize info service procedure, to reduce service cost and to add new source of revenue.

7. REFERENCES

- [1]. Chen qing, Qiao gangzhu, Zeng jianchao, Xu yubin, The Design and Implementation of Customer Relation Management System Based on .NET Architecture, CAID & CD ' 2003, Oct 18-20 2003, Hangzhou, China, pp.59-64.
- [2]. Wang HaoMing, The customer relation management and call center based on Internet, Journal of Information Technology, Oct 2002.pp. 39-41, (in Chinese).
- [3]. Li Jialu, What do the 400 million telephone users mean, Xinhua News Agency, Dec 18 2002, (in Chinese).
- [4]. Wang GuangYu, wireless application: The next" gold apple" of CRM, April 18 2003, <http://www.crmchina.com/web/>, (in Chinese).



Di Guoqiang is an associate Professor of Information management College, Jiangxi University of Finance & Economics. He graduated from Hehai University in 1992. He has published two books, over 20 Journal papers. His research interests are in CTI, e-commerce and m-commerce.

A Dynamic Data-Driven Application Simulation Framework *

Craig C. Douglas¹, Yalchin Efendiev²

¹University of Kentucky, Computer Science Department and Center for Computational Sciences
325 McVey Hall – CCS, Lexington, KY 40506-0045, USA and

Yale University, Computer Science Department
P.O. Box 208285, New Haven, CT 06520-8285, USA
Email: craig.douglas@yale.edu Tel.: +1-859-257-2326

²Texas A&M University, Mathematics Department
College Station, TX 77843-3368, USA
Email: yalchin.efendiev@math.tamu.edu Tel.: +1-979-845-1972

ABSTRACT

We describe, devise, and augment dynamic data-driven application simulations (DDDAS). DDDAS offers interesting computational and mathematically unsolved problems, such as, how do you analyze a generalized PDE when you do not know either where or what the boundary conditions are at any given moment in the simulation in advance? Only classical analysis works (sort of), but Sobolev theory definitely is missing. A summary of DDDAS features and why this is a really neat new field will be included in the talk. Two examples, contaminant tracking and wildfire modeling, will be used to motivate DDDAS.

Keywords: DDDAS, CFD, multiscale methods, automatic model changing, remote supercomputing and steering.

1. INTRODUCTION

In recent years, immense computing power has become available at the national and international supercomputer centers and local clusters of fast PCs. We also have had a proliferation of data acquisition and generation through the deployment of sophisticated new generations of sensors. The lack of coordination between current computational capacity and sensor technology impairs our ability to effectively utilize the flood of information available. This is a substantial barrier to achieving the potential benefit computational science can deliver to many application areas including contaminant tracking, wildfire modeling, transportation optimization, and many other fields.

We have identified four relatively diverse areas that have common issues that must be addressed for dynamic data driven application simulation (DDDAS) informational and computational sciences to have the promised impact toward addressing important problems.

These issues include:

- Effectively assimilating continuous streams of data into running simulations. These data streams most often will be
- Noisy data, but with known statistics:
 - Received from a large number of scattered remote locations and must therefore be assimilated to a usable computational grid.
 - Missing bits or transmission packets, as for example is the case in wireless transmissions.

- Injecting dynamic and unexpected data input into the model.
- Limited to providing information only at specific scales, specific to each sensor type.
- Warm restarts of simulations by possible incorporation of new data into parallel or distributed computations. Such processes are sensitive to communication speeds and data quality.
- Tracking and steering of remote distributed simulations to efficiently interact with the computations and to collaborate with other researchers.
- Components to assist researchers in their interpretation and analysis of collections of simulations. This will include designing and creating an application program interface and middleware.

Sensors and data generating devices may take many forms including other running computational simulations. The intent of this paper is to directly address several DDDAS enabling technologies in the context of a specific application area in order to provide techniques and tools to effectively demonstrate the potential of dynamic data driven simulations for other areas.

The primary application is contaminant tracking, which in groundwater reservoirs is modeled by strongly coupled systems of convection-reaction-diffusion equations. The solution process of such systems becomes more complicated when modeling remediation and clean-up technologies since they exhibit strong nonlinearities and local behavior. For efficient solution of this class of problems we need: (a) accurate, fast, and locally conservative approximation methods and (b) parallel adaptive methods that are dynamic in time. We shall solve these challenge problems using distributed computer systems (discussed above) and the latest developments in Eulerian-Lagrange localized adjoint method, discontinuous Galerkin method and/or streamline diffusion method in concert with domain decomposition and adaptive grid refinement techniques.

Many applications are essentially computer models that solve nonlinear, unsteady, coupled, partial differential equations. All require consistent initial conditions, adequate forcing fields, and boundary conditions to advance the solution in time. Collectively these fields represent the input data necessary to run the models. The input data can originate from observations, e.g., sensor based telemetry, can be internally generated from ensemble type simulations, or can be externally generated (e.g., providing boundary conditions for very high resolution regional models). The skill of these models to adequately represent realistic conditions is

* This research has been supported in part by the National Science Foundations under grants EIA-0219627, EIA-0218229, and EIA-0218721.

intimately tied to the quality, spatial and temporal coverage, and intelligent use of their input data sets. These applications in turn generate large amounts of output data that must be either analyzed or passed on to other more specialized subcomponents.

The traditional operating mode for most CFD applications is a static initialization with fixed forcing and boundary conditions followed by a limited exploration of the parameter space. This is clearly inadequate for many long term simulations, particularly when advances in observations capabilities, data assimilation techniques, and computers and networking can be leveraged to determine an optimal enough set of parameters needed for accurate and realistic forecasts.

DDDAS endows applications with dynamic data input capabilities by coupling the model and algorithms to the observations. The ultimate aim is to leverage the current state of the art in computing, networking, and observational instruments to produce a more realistic and accurate depiction of the state of a system than can be derived using either model or observations alone. We stress the fact that continuous data streams from observational instruments and sensors call for a radical change in model philosophy from static to dynamic data input.

Several hurdles stand in the way of achieving such an integrated, dynamically driven modeling system. These hurdles can be classified as data quality and management, computing and networking power, data assimilation and modeling algorithms, visualization, and hardware requirements.

Consider two justifications for doing DDDAS, which is distinctly more complicated than standard simulation methodologies. Many applications (e.g., 7 day weather forecasting) run fast enough already on parallel supercomputers. Starting a new simulation every time new data is available is then reasonable. However, much longer forecasts can be achieved when new data that is used to self-correct errors in predictions and automatically rescale for interesting features.

Some situations warrant a different approach. Major disaster simulation in real time is one. Suppose sensors can be placed and data collected quickly. Having access to a large parallel supercomputer is not a given. A set of WiFi connected laptops is much more likely. While current laptops have the computing power of a 1990 vintage Cray processor, this is insufficient for the simulations envisioned in this paper. Parallel calculations will have to cope with laptops dropping out of the communications network, returning unexpectedly, and how to distribute data as it arrives from sensors. Even how data can be assimilated and how much is needed or usable are open questions that must be addressed.

2. DDDAS CONTAMINANT TRACKING

For the software development we use new or improved modules and interfaces of SCIRun [3, 17, 21, 24] to implement various numerical methods for porous media flows. We now have several simulators that work both for rectangular as well as on general three dimensional unstructured grids. A finite volume element framework is utilized since this method maintains numerical conservation of

flux. Eventually we will use the mesh template mechanism from SCIRun, but additional basis function types and finite elements must be written for SCIRun.

Our first application is a single component contaminant transport in heterogeneous porous media taking into account convection and diffusion effects. Over time, this simple model will be further extended through the incorporation of additional physical effects as well as uncertainties.

The point of this example is to capture the effects of the heterogeneities. We have used various heterogeneous permeability fields in our simulations. These fields are generated using GSLIB libraries [4]. The heterogeneities are typically chosen to have large correlation features in the horizontal direction. Due to the presence of these heterogeneous features we expect irregular flow behavior, e.g., the contaminant can be transported faster in some regions while slower in others.

The mesh generator NETGEN [23] is used to discretize a given domain into a collection of tetrahedral. We wrote a C++ code that serves as an interface between NETGEN and SCIRun so that all mesh information required by the finite volume element algorithm can be accessed conveniently through this interface.

We also wrote a code to solve a time-dependent transport equation on the same grid setting. The flux computed from a much older, static data driven code is used as one of the inputs through virtual telemetry generated at another site. An upwind scheme is applied to resolve the transport part, which is then combined with an explicit time integration to obtain the transport quantity at the next time level.

The mathematical formulation of the problem is given by coupled equations that describe pressure distribution and the transport equations. The pressure field is described by the elliptic (or parabolic, in the presence of compressibility) equation and the transport of components is described by a convection-diffusion equation that is dominated by convection effects.

Advanced computational tools, methods, and algorithms for porous media flows, directly related to DDDAS, have also been of major research interest and efforts. We have worked in two main directions, namely, computational techniques and multiscale methods related to multi-component porous media flows and adaptive methods for general transport and diffusion equations.

One of our approaches is the use of multiscale interpolation techniques to map the sensor data from sparse locations into the solution space during simulations. The interpolation operator is built for general nonlinear parabolic equations that include various porous media processes. Moreover, we take advantage of the interpolation operator and use multiscale numerical methods for the problem. These methods are significantly faster than single scale methods. We are testing our method on a variety of synthetic examples. In particular we can show that frequent updating of the sensor data in the simulations significantly improves the prediction results. The frequency of sensor data updating in the simulations is related to the streaming capabilities.

Data that is transmitted through a telecommunications system

is commonly referred to as telemetry. The transmission media can be one or more of land lines, underwater lines, microwave, or satellite based. There is both latency and broadcast time based on distance and resistance in the physical media that determines how long the data takes to get to the receiver.

Real telemetry used in predictive contaminant monitoring comes in small packets from sensors in wells or placed in an open body of water. There may be a few sensors or many. With virtual telemetry, we can trivially vary the amount of telemetry that we sample and its frequency.

Real telemetry based on high resolution photographs from space on a slow space to land transmission system can be quite a challenge. However, we are not dealing with this situation presently.

Real telemetry is usually expensive to receive (if it is even available on a long term basis), tends to be messy, comes in no particular order, and can be incomplete or erroneous due to transmission problems or sensor malfunction. For predictive contaminant telemetry, there are added problems that due to pesky legal reasons (corporation X does not want it known that it is poisoning the well water of a town), the actual data streams are not available to researchers, even ones providing the simulation software that will do the tracking.

Virtual telemetry has the advantage that it is inexpensive to produce from real time simulations and can easily be transmitted using modified forms of open source streaming software.

We will generate multiple streams continuously for extended periods (e.g., months or years): clean data, somewhat error prone data, and quite lossy or inaccurate data. By studying all of the streams at once we will be able to devise DDDAS components useful in predictive contaminant modeling.

The basic technique that we are using is to take an old, robust, formerly state of the art, but different model. (A different model guarantees different errors in predictions.) The corresponding code uses only static input data and makes long term predictions with the capability of getting the transient data throughout the simulation. Instead of trying to run the old code very, very fast, we run it in real time using small time steps for all components. We run the old code on either a fast, cheap PC or a small parallel computer depending on how much data we wish to generate per time step. The code sleeps most of the time while waiting for the wall clock to catch up. Our sample time steps range from one minute to a few hours.

The old code uses a 3D tensor product mesh with finite differences. The code is conservative, which is essential in the situation we are interested in. We use data from a small subset of computed values and assume that a sensor is placed there. The location (or some unique identification), time, and a few pieces of floating point data are all that have to be transmitted on a per sensor time dependent basis.

Broadcasting the telemetry as audio has the advantage that there are many programs to choose from to generate the data streams and to "listen" to them on the receiving end.

Broadcasting the telemetry as a movie stream or a full 3D

visualization has the advantage that it can be trivially visualized on the receiving end. This is particularly attractive when studying incomplete and/or erroneous data streams.

Eventually, we decided to place MP3 data headers around the telemetry data. The actual header information is contained in Table 1. A small program was written that takes sample data and produces an output file containing the same data only with MP3 headers placed where they need to be. This is the basis of the CH3 encoding we have developed.

We decided that we had to use Open Source software since we determined during the project that we would have to modify both the streaming code and the receiving client in order to eliminate unwanted data compression and filtering techniques aimed at integer data.

MP3 normally uses Huffman encoding. Many encoders also implement filters based on knowing which audio or visual frequencies humans cannot hear or see, respectively. Since the data is integer based, the values out of range are zeroed out. Imagine a floating point number whose exponent has been zeroed (e.g., 1.2×10^{31} might become just 1.2, a small, but rather noticeable error in data transmission).

We are using a modified version of the Gini streaming server [22]. Gini must be modified since it checks the data to see if it corresponds to legitimate MP3 audio data. Our Ch3 format data fails this criterion. The fix is to comment out a few lines of code in one file (mp3.c) of the streamer.

Receiving the data as audio is a function of modifying an Open Source MP3 client like XMMS [1]. We added the CH3 format to produce a new version of XMMS which we call xccs. The xccs code for the receiver has been integrated into a multi-threaded SCIRun module called "StreamReader." This module is part of the DDDAS package in SCIRun.

As a preliminary step, we first implemented a DDDAS module called Reader that is capable of parsing a flat file containing 3D tensor product mesh data and visualizing the data as either a 3D "LatVol" mesh (the LatVol mesh is one of the basic SCIRun Fields) or a basic "PointCloud" mesh (specific for unstructured data). This step separates the problem of accurately parsing and visualization from the problem of receiving data from a stream.

From the perspective of the Reader and StreamReader modules, the 3D tensor product mesh data is a set of n cell-centered solution points that represent values at nodes in a 3D structured grid. As mentioned earlier, SCIRun represents this type of grid as a LatVol mesh. Since the incoming mesh data always has a header indicating the total number of solution points in the mesh, the dimensions of the resulting 3D structured grid can be derived from the header. Once the header is parsed, the Reader module reads the appropriate number of floats (solution points) in and fills the LatVol mesh with them. The LatVol is then visualized with existing SCIRun modules.

Additionally, the ability to continually update the LatVol (or PointCloud) mesh by reading in data files from sequential time steps is a feature of the visualization capabilities of the Reader module, allowing the the simulation of continual updates from a data stream.

The next step was to couple the parsing and visualization capability with the xccs stream receiver code mentioned above. The result was a SCIRun module called StreamReader that continually reads data from a stream on a remote machine, parses the contents, and produces a viewable LatVol mesh from the solution points contained in the data. The StreamReader module is multithreaded with two threads, one that reads and caches away the data, and another that waits for a complete mesh, processes it, and sends it downstream to be processed. There is some data loss between reads because of both network latency and processing that necessarily occurs between reads. Currently, this data loss is ignored since the solution headers are relatively infrequent and lack the information to determine which chunks of data have been lost. Hence, the LatVol meshes produced generally have some degree of error and this shows up as nodes in the mesh being shifted from their correct position.

We have found that in order to get complete meshes, the incoming CH3 needs to be modified to include intermediate headers in the mesh data. This way we can eventually get complete meshes even if some stream data is lost. This change will most likely be a natural outgrowth of development with more realistic sensor data.

The StreamReader module is designed to be extensible with the ability to accommodate different formats of streamed data. A new form using XML is being developed presently.

The StreamReader module is easily coupled to the adaptive mesh refinement modules in SCIRun, so that quite general meshes and finite elements are readily available in constructing DDDAS applications quickly and robustly.

A new, highly sophisticated DDDAS enabled code is being completed. The legacy finite difference code is broadcasting several streams of data, which are received by the StreamReader module in SCIRun.

We are developing a comprehensive sensor data format, which we hope will work with real, digital sensor data as well as our own virtual telemetry. We include mechanisms to identify sensors by an id tag, GPS, and/or GIS information. Quite general data can be sent.

For our current application, very simple data is sent from fixed sensors. Hence, we only need to identify the sensor and provide its data. Using 25 wells randomly scattered across the domain, we can broadcast data from 1000 sensors in less than six seconds using a very general sensor format. Using a minimal sensor format allows a transmission time well under six seconds.

Data is interpolated onto the new code's mesh using a SCIRun module. The finite element module in SCIRun has been modified to allow much more general finite elements. We are developing techniques to inject the telemetry data into the simulation in a manner that is non-intrusive when error analysis indicates that the simulation is already within error tolerances. Theoretical tools for both linear and nonlinear problems are being investigated.

Unlike data assimilation methods, there is no reason to inject telemetry that will have no real bearing on the accuracy of previous predictions. Only when the predictions are provably outside of error bounds in a region of the domain do

we have to inject the telemetry at some time step of the simulation. Once telemetry is injected, we have to determine if a warm restart is required or if solving a correction problem will allow continuing the simulation.

3. CONCLUDING REMARKS

In [16], we have developed a new multiscale computational approach for multi-phase flow that is applicable for multi-component single phase flow. The latter is currently under investigation, which will be further extended to a multi-phase multi-component scenario. In [16], multiscale finite volume techniques along with a perturbation argument are used to upscale porous media flows, such as single-phase and two-phase immiscible flows, Richards' equations. In [15], analysis of our multiscale finite volume element method is presented.

The papers [9-12] are dedicated to the construction and analysis of novel multiscale methods for nonlinear elliptic and parabolic equations. In these papers we systematically investigate our new approach showing that it can handle various nonlinearities in the homogenization process. The applications of this work to single and two-phase flows are presented in [7, 8], where we have proposed a generalized convection-diffusion approach for up-scaling porous media flows. The methods are tested successfully for both single and two phase flows in heterogeneous porous media.

In [2, 18] we have developed, theoretically studied, implemented, and tested adaptive finite volume methods for transport equations in general domains covered by unstructured tetrahedral meshes. We have established the reliability and the efficiency of the schemes and tested them on computing the concentration of passive chemicals in heterogeneous aquifers.

In [13, 14] the authors use ELLAM technique for accurate simulation of convection dominated problems. In particular, wavelets are employed as basis functions. The mathematical analysis of the method is presented in [20]. In [19] the authors investigate strategies for parallel computing of the black oil model.

In [5] we describe our initial views of how to take an application that uses static, initial data and use it to generate data in a manner similar to real data generated by sensors in the field, so-called virtual telemetry, for new DDDAS enabled codes. This provided a framework for all three groups to produce the new codes that use telemetry-like data for injection into the data streams. In [6] we justify virtual telemetry and provide more details of how we transmit data using a data format that is similar to MP3.

4. REFERENCES

- [1] P. Alm, T. Nilsson, O. Hallnas, and H. Kvalen, (2003). XMMS - X multimedia system: A cross platform multimedia player. <http://www.xmms.org>.
- [2] C. Carstensen, R. Lazarov, and S. Tomov, (2003). Explicit and averaging a posteriori error estimates for adaptive finite volume methods. *Preprint, Isaac Newton Institute of Mathematical Sciences*. Available at <http://www.newton.cam.ac.uk/preprints/NI03010.pdf>.

- [3] M. Cole, and S. Parker, (2001). Dynamic compilation of C++ template code. In *Proceedings of the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA'01)*.
- [4] C.V. Deutsch and A.G. Journel, (1998). *GSLIB: Geostatistical software library and user's guide, 2nd edition*. Oxford University Press, New York.
- [5] C.C. Douglas, Y. Efendiev, R. Ewing, R. Lazarov, M.R. Cole, C.R. Johnson, and G. Jones, (2003a). Virtual telemetry middleware for DDAS. In *Computational Sciences - ICCS 2003*, 4:279-288.
- [6] C.C. Douglas, C.E. Shannon, Y. Efendiev, R. Ewing, V. Ginting, R. Lazarov, M.J. Cole, G. Jones, C.R. Johnson, and J. Simpson, (2003). A note on dynamic data driven application simulations using virtual telemetry. In *Proceedings of ICSPACE 2003*, pages 193-198.
- [7] Y. Efendiev and L. Durlofsky, Accurate subgrid models for two-phase flow in heterogeneous reservoirs. SPE 79680, presented at the SPE Reservoir Simulation Symposium held in Houston, Texas, February 3-5, 2003.
- [8] Y. Efendiev and L. Durlofsky, (2003). Generalized convection-diffusion model for subgrid transport in porous media. *SIAM Multiscale Modeling Simulation*, 1:504-526.
- [9] Y. Efendiev and A. Pankov. Homogenization of nonlinear random parabolic operators. Submitted to *EJDE*. Available at <http://www.math.tamu.edu/~yalchin.efendiev/ep-hom-parab.ps>.
- [10] Efendiev, Y. and Pankov, A. Meyers type estimates for approximate solutions of nonlinear elliptic equations and their applications. Submitted to *Numer. Math.*
- [11] Y. Efendiev and A. Pankov. Numerical homogenization of nonlinear random parabolic operators. To appear in *SIAM Multiscale Modeling Simulation*.
- [12] Y. Efendiev and A. Pankov. (2004). Numerical homogenization of monotone operators. *SIAM Multiscale Modeling Simulation*, 2:62-79.
- [13] R.E. Ewing, J. Liu, and H. Wang. Adaptive biorthogonal spline schemes for advection-reaction equations. In press in *J. Comput. Phys.*
- [14] R.E. Ewing, J. Liu, and H. Wang. (2003). Adaptive wavelet method for advection-reaction equations. *Contemporary Math.*, 329:119-130.
- [15] V. Ginting. Analysis of two-scale finite volume element for elliptic problem. To be submitted.
- [16] V. Ginting, R.E. Ewing, Y. Efendiev, and R. Lazarov. Upscaled modeling for multiphase flow. To appear in *J. Comput. Appl. Math.*.
- [17] C.R. Johnson, S. Parker, D. Weinstein, and S. Heffernan (2002). Component-based problem solving environments for large-scale scientific computing. *J. Concurrency and Computation: Practice and Experience*, 14:1337-1349.
- [18] R. Lazarov and S. Tomov (2002). A posteriori error estimates for finite volume element approximations of convection-diffusion-reaction equations. *Comput. Geosci.*, 6:483-503.
- [19] J. Liu, Z. Chen, R.E. Ewing, G. Huan, B. Li, and Z. Wang (2003). Parallel computing in the black oil model. *Contemporary Math.*, 329:253-262.
- [20] J. Liu, B. Popov, H. Wang, and R.E. Ewing. Convergence analysis of wavelet schemes for convection-reaction equations under minimal regularity assumptions. Submitted to *SIAM J. Numer. Anal.*
- [21] M. Miller, C. Moulding, J. Dongarra, and C. Johnson. (2001). Grid-enabling problem solving environments: A case study of SCIRun and NetSolv. In *Proceedings of the High Performance Computing Symposium (HPC 2001)*, 2001 Advanced Simulation Technologies Conference, pages 98-103. Society for Modeling and Simulation International.
- [22] K. Pifko, B. Dakay, and T. Szerb (2003). Gini. <http://gini.sourceforge.net>.
- [23] J. Schoeberl (1997). Netgen - an advancing front 2D/3D-mesh generator based on abstract rules. *Comput. Visual. Sci.*, 1:41-52.
- [24] SCIRun (2002). SCIRun: A Scientific Computing Problem Solving Environment. Scientific Computing and Imaging Institute (SCI). Available at <http://software.sci.utah.edu/scirun.html>.
- [25] Vodacek, A. (2001). Wildland fire needs assessment workshop report. Available at <http://www.cis.rit.edu/research/dirs/research/fires.html>.
- [26] W. Zhao (2001). Challenges in design and implementation of middlewares for real-time systems, *J. of Real-Time Systems*, 20:1-2.

A Hybrid Method for the Update of Sub-domain Interfaces *

Q. P. Guo¹ and C.-H. Lai²

¹ School of Computer Science and Technology

Wuhan University of Technology, Yujiatou Campus, Wuhan 430063, China. Email: qpguo@public.wh.hb.cn

² School of Computing and Mathematical Sciences, University of Greenwich, Old Royal Naval College,
Park Row, Greenwich, London SE10 9LS, UK. Email: C.H.Lai@gre.ac.uk

ABSTRACT

Many engineering applications involve significant computational time, which needs to be reduced by means of a fast solution method or parallel and high performance algorithms. It is well known that multigrid methods serve as a fast iterative scheme for linear and nonlinear diffusion problems. On the other hand, many engineering applications require collaborative work using components, which are being solved in a distributed environment. Therefore it is necessary to develop collaborative methodologies in modern distributed computing environment. This paper develops a hybrid method for the update of sub-domain interfaces in an attempt to provide a distributed algorithm suitable for future Grid computing.

1. INTRODUCTION

Many engineering applications involve significant computational time, which needs to be reduced by means of a fast solution method or parallel and high performance algorithms. It is well known that multigrid methods serve as a fast iterative scheme for linear and nonlinear diffusion problems. There are two major issues in this fast iterative solver. First, multigrid methods are usually very difficult to parallelise and the performance of the resulting algorithms are machine dependent. Early work in parallelisation of multigrid methods is due to Barkai and Brandt [1], Chan [2], Frederickson [3], Naik [4]. Methods developed by these authors concerned the load balancing between processors and the full use of all co-existing coarse level meshes in order to fit into the parallelism requirement.

With the development of the Grid [5], a modern concept of computing environment, distributed resources such as workstations and PC's amongst other data storage facilities may be shared to facilitate ever larger or independent simulations being done at different geographical locations. One typical application being engaged at Greenwich is electronic packaging techniques. Such applications often involve the combination of a large number of components designed by engineers from different geographical locations. These components are essentially equivalent to the subdomains, and the combination of these components amounts to a kind of coupling in a domain decomposition context. Another typical application involves the coupling of a few different physical components describing viscous flow, inviscid flow, compressible potential flow and potential flow [6]. Typical techniques involved are the embedding of one mesh into another one essentially directed to

different mathematical problems for achieving improved local accuracy. In general these components may be generated by means of computer-aided engineering tools at different office locations across a region. Finally applications interest the first author is new parallel multi-grid algorithms for heat transfer problems in ceramic/metal composites [8].

This paper follows the idea of a non-overlapped domain decomposition method, which generates interior boundaries as the interface of sub-domains, and a multi-level methods being employed in each of the sub-domains. The hybrid term comes in the treatment of the interior boundaries using alternatively an iterative scheme and a coarse level extrapolation of previous iterative approximations along the interior boundaries. Numerical experiments are presented of the communications-saving between subdomains and speed up of the hybrid scheme.

2. NON-OVERLAPPED DOMAIN DECOMPOSITION METHODS

The idea of domain decomposition has a long evolving history. Many literatures may be found, for examples [9], [10] and [11]. Domain decomposition involves the subdivision of a given problem into a number of subproblems. Each of these subproblems can be solved separately before being combined to give the global solution of the original problem. The subdivision can be done at either the physical problem level or the discretised problem level. At the discretised problem level, the resulting linear system of equations is rearranged as a collection of smaller linear systems which may be solved independently. At the physical problem level, regions governed by different mathematical models or different material properties are identified and decomposed into different subdomains resulting in a number of locally regular subproblems. It should be noted at this stage that the use of a distributed environment is highly suitable for this class of methods.

This paper concentrates on applying the domain decomposition approach directly at the physical problem level. The meshing and the discretisation of the continuous problem in a subdomain are carried out individually. The subdomains intersect to form interior boundary known as interface, to which solutions are required in order to solve each of the non-overlapped subproblems. Therefore solution methods for solving these interface problems are needed. This paper does not deal with the overlapped subdomains, where updating of the boundary data of the subdomains may be achieved at the expense of having to solve some larger problems defined in the subdomain [12].

* Supported by NSFC (Grant No.: 60173046) and NSF of Hubei Province (Grand No.: 2000J153)

The non-overlapped approach has an advantage in modular software design involving the use of Grid computing where collaboration between organisations located at different geographical locations may be made possible. There is also another advantage of the approach in the coupling of different mathematical models within the entire physical domain and enabling the best model and numerical schemes are being used in each of the subdomains [13] [14].

Consider the boundary value problem defined by

$$\begin{cases} Lu = f & \text{in } \Omega \\ Bu = g & \text{on } \partial\Omega \end{cases} \quad (1)$$

where L is linear differential operator. The domain Ω is partitioned into S non-overlapped subdomains Ω_i , $i = 1, L, S$ such that $\cup \Omega_i = \Omega$ and $\cap \Omega_i = \emptyset$, Denoting $\gamma = (\cup \partial\Omega_i) \setminus \partial\Omega$ the interior boundary or interface and $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ the common interface between two adjoining subdomains Ω_i and Ω_j . The sub problem is defined by

$$\begin{cases} Lu_i = f & \text{in } \Omega_i \\ \Phi(u_i) = \Phi(u_j) & \text{on } \Gamma_{ij} \\ \Psi(u_i) = \Psi(u_j) & \text{on } \Gamma_{ij} \end{cases}$$

where Φ and Ψ are interface conditions depending on the differential operator L . For second order partial differential equations $\Phi(v) = v$ and $\Psi(v) = \frac{\partial v}{\partial n}$. A general iterative procedure may be used to obtain solutions to the interface equations by using suitable weighted averaging of Φ and Ψ along the interface Γ_{ij} , after each iteration, such as

$$\bar{\Phi}_i = \alpha \Phi(u_i^{(k)}) + (1 - \alpha) \Phi(u_j^{(k)}) \quad \text{on } \Gamma_{ij} \quad (2a)$$

$$\bar{\Psi}_i = \beta \Psi(u_i^{(k+1/2)}) + (1 - \beta) \Psi(u_j^{(k+1/2)}) \quad \text{on } \Gamma_{ij} \quad (2b)$$

where α and β are weightings. It is obvious that

$$\bar{\Phi}_i = \Phi(u_i^{(k)}) \quad \text{on } \partial\Omega_i \cap \partial\Omega \quad (3a)$$

$$\bar{\Psi}_i = \Psi(u_i^{(k+1/2)}) \quad \text{on } \partial\Omega_i \cap \partial\Omega \quad (3b)$$

Therefore the general iterative procedure involves two steps as given below,

Algorithm 1: Iterative non-overlapped subdomains.

```

Do
  Parallel Do  $i = 1$  to  $S$ 
  Solve:-
    
$$\begin{cases} Lu_i^{(k+1/2)} = f & \text{in } \Omega_i \\ \Phi(u_i^{(k+1/2)}) = \bar{\Phi}_i & \text{on } \partial\Omega_i \end{cases} \quad (4)$$


```

End Parallel Do

Parallel Do $i = 1$ to S

Solve:-

$$\begin{cases} Lu_i^{(k+1)} = f & \text{in } \Omega_i \\ \Psi(u_i^{(k+1)}) = \bar{\Psi}_i & \text{on } \partial\Omega_i \end{cases} \quad (5)$$

End Parallel Do

Until converged

There above scheme is a multi-domain Dirichlet-Neumann method and has two interpretation [11]. First the finite element implementation of the scheme leads to the substructuring iterative procedure. Second it is the preconditioner being used in the framework of any iterative method to solve the Steklov-Poincare equation. The main disadvantage of the algorithm is that it would require solving two times each of the subproblems in each of the iterations. On the other hand the convergence rate of the above scheme deteriorates when the number of subdomains becomes large. This is simply due to the slow propagation of boundary information only through the exchange of information between neighbouring subdomains.

3. A HYBRID INTERFACE UPDATE METHOD

One idea to overcome the slow information propagation in the iterative scheme described in Section 2 is to use a coarse global problem set over the entire domain in order to have a mechanism of global information propagation amongst all of the subdomains [11]. However the second parallel do loop as shown in the two-step iterative algorithm in Section 2 still exist. The other major issue of a coarse global problem on the entire domain is that it is intrinsic sequential. Therefore it is difficult to parallelise the coarse global problem.

The idea that would be exploited in this paper is to replace the second parallel do loop by means of a global information movement. It has the same advantage of employing a smoothing process as that on the fine grid of a multigrid method. The idea leads to a hybrid method consists of a parallel do loop and a smoothing process instead of two parallel do loop. This algorithm is described as below.

Algorithm 2: A hybrid algorithm for non-overlapped subdomains.

Do

Parallel Do $i = 1$ to S

Solve:-

$$\begin{cases} Lu_i^{(k+1/2)} = f & \text{in } \Omega_i \\ \Phi(u_i^{(k+1/2)}) = \bar{\Phi}_i & \text{on } \partial\Omega_i \end{cases} \quad (6)$$

End Parallel Do

Global smoothing:-

$$\begin{cases} Lu^{(k+1)} = f & \text{in } \Omega \\ Bu^{(k+1)} = g & \text{on } \partial\Omega \end{cases} \quad (7)$$

Until converged

The global smoothing can be one or two steps of Gauss-Seidel iterations. This smoothing process will make sure the interface condition Ψ is always updated. A local multigrid method can be applied to eqn (6).

4. INTERFACE UPDATE WITH VIRTUAL BOUNDARY FORECAST METHOD

4.1 Interface update with time dimension and space information

It is observed that many interfaces or virtual boundaries are created in the domain decomposition. If unknowns on the virtual boundaries could be predetermined or given as they should be, the whole problem would be decomposed as several independent sub-domain problems.

Although it is impossible to predetermine the interface values on each subdomain, but if the process of approaching the real values on the interface could be accelerated, a speedup should be gained in the above parallel algorithm.

In the *algorithm 2*, the interface update highly depends on the Gauss-Seidel iterations for propagating values of given external real boundary to internal virtual boundaries as well as to other internal points. That is a kind of space-based information propagation. If unknown space is huge, this kind of information propagation would very slow.

In fact the interface update to approach the final solution is a time sequence for each unknowns on the virtual boundary. If the real values of the virtual boundaries could be in some extent precisely predicted by the previous values, it should speedup problem solving. From this consideration the author proposes a virtual boundary forecast method (VBF), which extrapolates unknowns on the interface with a few last historical iteration values [15]. Combining the VBF method the *algorithm 2* is modified as below.

Algorithm 3: A hybrid algorithm with VBF for non-overlapped subdomains.

```

Do
  Parallel Do  $i = 1$  to  $S$ 
    Virtual boundary forecast on  $\partial\Omega_i$  :
       $u^{(K+1)} = \text{Forecast}(u^{(K)}, u^{(K-1)}, u^{(K-2)})$  (8)
    Solve:-
      
$$\begin{cases} Lu_i^{(k+1/2)} = f & \text{in } \Omega_i \\ \Phi(u_i^{(k+1/2)}) = \bar{\Phi}_i & \text{on } \partial\Omega_i \end{cases}$$
 (9)
    End Parallel Do

    Global smoothing:-
      
$$\begin{cases} Lu^{(k+1)} = f & \text{in } \Omega \\ Bu^{(k+1)} = g & \text{on } \partial\Omega \end{cases}$$
 (10)
  Until converged

```

To create the initial values on the virtual boundaries, i.e. the $u^{(k)}$, $u^{(k-1)}$, $u^{(k-2)}$, let all processors in parallel in Ω with the given real boundary $\partial\Omega$ do the same Gauss-Seidel iteration three times on a

very coarse grid, the grid size of which is determined by the whole length of a dimension divided with the number of involved processors. Then each processor chooses its own $u^{(k)}$, $u^{(k-1)}$, $u^{(k-2)}$ of coarse grid on $\partial\Omega_i$ as initial values of its own virtual boundaries. Interpolating and then extrapolating can obtain the finest grid values on the virtual boundaries.

4.2 A full multigrid approach with the virtual boundary forecast

A bottleneck in the *algorithm 3* is the virtual boundary calculation. It is obvious that interpolating very coarse grid values to get the finest grid values for the interface should bring larger errors, therefore slow down the solution. To avoid that a full multigrid approach could be adopted here, giving an algorithm as following.

Algorithm 4: A hybrid algorithm with VBF and full multigrid approach for non-overlapped subdomains.

```

Parallel Do  $i = 1$  to  $S$ 
  Full multigrid calculation in  $\Omega$  on  $\partial\Omega$ 
  Until the initial grid size
    to get the initial subdomain boundary values (11)
End Parallel Do
Do
  Parallel Do  $i = 1$  to  $S$ 
    Virtual boundary forecast on  $\partial\Omega_i$  :
       $u^{(K+1)} = \text{Forecast}(u^{(K)}, u^{(K-1)}, u^{(K-2)})$  (12)
    Solve:-
      
$$\begin{cases} Lu_i^{(k+1/2)} = f & \text{in } \Omega_i \\ \Phi(u_i^{(k+1/2)}) = \bar{\Phi}_i & \text{on } \partial\Omega_i \end{cases}$$
 (13)
    End Parallel Do
    Global smoothing:-
      
$$\begin{cases} Lu^{(k+1)} = f & \text{in } \Omega \\ Bu^{(k+1)} = g & \text{on } \partial\Omega \end{cases}$$
 (14)
  Until converged

```

The initial grid size in *algorithm 4* depends on the number of processors involved. It is defined as a length of one dimension, from which the whole domain is sliced, divided by the number of processors. Strictly obeying this definition, it is guaranteed that only the initial virtual boundary values of the subdomains are calculated. Then each processor chooses its own $u^{(k)}$, $u^{(k-1)}$, $u^{(k-2)}$ as initial values of its own virtual boundaries.

The meaning of equation (12) in the *algorithm 4* is different with equation (8) in the *algorithm 3*. The sequence $u^{(k-2)}$, $u^{(k-1)}$, $u^{(k)}$ is not only a time sequence, but also a from-coarser-to-finer grid space sequence as well.

The equations (13) and (14) are solved on from the initial grid size to the finest grid size until converged. Initial values of equation (13) on next finer grid can be obtained from the last coarser grid by interpolation. Since there is no necessary to get accurate solution of equation (13) until on finest grid, fewer-levels of multi grid method with fewer cycles is applied to solve equation (13) on each grid level, there is no communication between neighbour processors. Based on the same consideration less iteration is needed for solving equation (14), the communication should occur in this operation.

4.3 Virtual boundary forecast methods

Concrete forecast methods of the virtual boundary are various. In this section several forecast approaches are analysed.

4.3.1 A linear VBF method

A linear forecast method is defined as below:

$$u^{(S)} = u^{(K)} + C_1(S - K) \quad (15)$$

Where the C_1 is a parameter waiting to be determined, the K is a repetition number of the algorithm (also the grid level in *algorithm 4*) and the S is a variable. It is obvious that span of the linear forecast depends on the slope C_1 . Two cases are analysed here, one belongs to forward span forecast; the other is remained with backward span forecast.

4.3.1.1 Forward span linear forecast

Let the S equals $K - 1$ in the equation (15), then the C_1 is determined as

$$C_1 = u^{(K)} - u^{(K-1)}$$

Therefore the linear forecast formula is

$$u^{(K+1)} = u^{(K)} + (u^{(K)} - u^{(K-1)}) \quad (16)$$

or $u^{(K+1)} = 2u^{(K)} - u^{(K-1)}$

If the forecast sequence $\{u^{(K)}\}$ is a non decreasing sequence and has a limit, the linear forecast formula (16) is a typical forward span linear forecast method.

4.3.1.2 Backward span linear forecast

Let the S equals $K+1$ and the $u^{(K+1)} = 0.5(u^{(K)} + u^{(K-1)})$ in the equation (15), then the C_1 can be determined as

$$C_1 = 0.5(u^{(K-1)} - u^{(K)})$$

So the linear forecast formula (15) becomes

$$u^{(S)} = u^{(K)} + 0.5(u^{(K-1)} - u^{(K)})(S - K) \quad (17)$$

If the forecast sequence $\{u^{(K)}\}$ is a non decreasing sequence and has a limit, a typical backward span linear forecast formula is

$$u^{(K+1)} = u^{(K)} + 0.5(u^{(K-1)} - u^{(K)}) \\ = 0.5(u^{(K)} + u^{(K-1)}) \quad (18)$$

The same conclusion is true for the formula (16) and (18) if the forecast sequence $\{u^{(K)}\}$ is a non increasing sequence and has a limit.

4.3.2 A quadratic VBF method

A quadratic forecast method is defined as:

$$u^{(S)} = u^{(K-1)} + C_1(S-K+1) + C_2(S-K+1)^2 \quad (19)$$

Here the C_1 and C_2 are parameters waiting to be determined, the K is a repetition number of the algorithm (also the grid level in *algorithm 4*) and the S is an integer variable. Let the S equals K and $K - 2$, the relations of C_1 and C_2 are exposed as

$$C_1 + C_2 = u^{(K)} - u^{(K-1)} \\ C_1 - C_2 = u^{(K-1)} - u^{(K-2)}$$

So the C_1 and C_2 can be determined as

$$C_1 = 0.5(u^{(K)} - u^{(K-2)}) \\ C_2 = 0.5(u^{(K)} - 2u^{(K-1)} + u^{(K-2)}) \quad (20)$$

Therefore the quadratic forecast formula can be deduced as

$$u^{(K+1)} = u^{(K-1)} + 2C_1 + 4C_2 \\ = u^{(K-1)} + u^{(K)} - u^{(K-2)} + 2(u^{(K)} - 2u^{(K-1)} + u^{(K-2)})$$

Hence the quadratic forecast formula is

$$u^{(K+1)} = u^{(K-2)} + 3(u^{(K)} - u^{(K-1)}) \quad (21)$$

4.3.3 A extremum quadratic VBF method

In order to accelerate the progress to approach real values of virtual boundaries, an extremum forecast could be applied in the quadratic VBF method. The extremum value of the equation (19) is

$$u_{\text{extremum}} = u^{(K-1)} - 0.25C_1^2/C_2 \quad (22)$$

And the extremum quadratic VBF method is defined as

$$u^{(K+1)} = u^{(K-1)} - 0.25C_1^2/C_2 \quad (23)$$

4.3.4 Forecast span comparison of the VBF methods

If the forecast sequence $\{u^{(K)}\}$ is a non decreasing (or increasing) sequence and has a limit, several theorems can be proved and used for forecast span comparisons of above VBF methods. In following sections, for simplicity, some notations are adopted, that is, denoting the value of forecast for the forward linear forecast VBF method with $u^{(K+1)}_{\text{forward-linear}}$, for the backward linear forecast VBF method with $u^{(K+1)}_{\text{backward-linear}}$, for the quadratic VBF method with $u^{(K+1)}_{\text{quadratic}}$, and for the extremum quadratic VBF method with $u^{(K+1)}_{\text{extrem-quadratic}}$.

Theorem 1. Span of the quadratic VBF method is not greater than the span of extremum quadratic VBF method.

Proof: It is true from the definitions of the two VBF methods.

Therefore the relation is true that

$$u^{(K+1)}_{\text{extrem-quadratic}} \geq u^{(K+1)}_{\text{quadratic}}$$

■

Theorem 2. Span of the quadratic VBF method is not greater than the span of forward linear forecast VBF method.

Proof: Comparing the equation (16) and (21) it can be obtained that

$$u^{(K+1)}_{\text{forward-linear}} - u^{(K+1)}_{\text{quadratic}} \\ = (2u^{(K)} - u^{(K-1)}) - (u^{(K-2)} + 3(u^{(K)} - u^{(K-1)})) \\ = (u^{(K-1)} - u^{(K-2)}) - (u^{(K)} - u^{(K-1)})$$

Since the forecast sequence $\{u^{(K)}\}$ is a non decreasing (or increasing) sequence, the above difference of spans have to obey that

$$u^{(K+1)}_{\text{forward-linear}} - u^{(K+1)}_{\text{quadratic}} \geq 0$$

so that

$$u^{(K+1)}_{\text{forward-linear}} \geq u^{(K+1)}_{\text{quadratic}}$$

■

Theorem 3. Span of the backward linear forecast VBF method is not greater than the span of the quadratic VBF method.

Proof: Observing the equation (18), the following relations should be obeyed, that is:

$$\text{Min}\{u^{(K-1)}, u^{(K)}\} \leq u^{(K+1)} \leq \text{Max}\{u^{(K-1)}, u^{(K)}\}$$

Therefore the forecast sequence $\{u^{(K)}\}$ is not a non decreasing (or increasing) sequence. If the forecast sequence $\{u^{(K)}\}$ is a non decreasing (or increasing) sequence, then from the equation (21) it can be observed that the forecast value $u^{(K+1)}_{\text{quadratic}}$ is increased (or decreased) steadily. So that following relation is true

$$u^{(K+1)}_{\text{backward-linear}} \leq u^{(K+1)}_{\text{quadratic}}$$

■

From theorem 1, 2 and 3, the following relations can be derived, that is

Theorem 4. The span relations of the four typical forecast methods are as below

$$u^{(K+1)}_{\text{backward-linear}} \leq u^{(K+1)}_{\text{quadratic}} \leq (u^{(K+1)}_{\text{forward-linear}} \cdot u^{(K+1)}_{\text{extrem-quadratic}})$$

4.3.5 Convergence of the VBF methods in the hybrid algorithm

The hybrid algorithm is inherently convergent as the Gauss-Seidel iteration, since it essentially is Gauss-Seidel iterations, and the entire parallel do as well as the VBF method are used for offering better initial values for the Gauss-Seidel iterations to accelerate their convergence. So that the iteration sequence $\{u^{(K)}\}$ on the interface is convergent. Based on this judgement it is easy to prove that above four forecast approaches are convergent.

Denote the limit of $\{u^{(K)}\}$ as $u^{(Real)}$. It is obvious that:

$$\begin{aligned} \lim_{k \rightarrow \infty} u^{(K+1)}_{\text{extrem-quadratic}} &= \lim_{k \rightarrow \infty} (u^{(K-1)} - 0.25C_1^2/C_2) \\ &= u^{(Real)} - 0.25(u^{(Real)} - u^{(Real)})^2 / (u^{(Real)} - u^{(Real)}) \\ &= u^{(Real)} \end{aligned}$$

$$\begin{aligned} \lim_{k \rightarrow \infty} u^{(K+1)}_{\text{forward-linear}} &= \lim_{k \rightarrow \infty} (2u^{(K)} - u^{(K-1)}) \\ &= 2u^{(Real)} - u^{(Real)} = u^{(Real)} \end{aligned}$$

$$\begin{aligned} \lim_{k \rightarrow \infty} u^{(K+1)}_{\text{quadratic}} &= \lim_{k \rightarrow \infty} (u^{(K-2)} + 3(u^{(K)} - u^{(K-1)})) \\ &= u^{(Real)} + 3(u^{(Real)} - u^{(Real)}) = u^{(Real)} \end{aligned}$$

$$\begin{aligned} \lim_{k \rightarrow \infty} u^{(K+1)}_{\text{backward-linear}} &= \lim_{k \rightarrow \infty} 0.5(u^{(K)} + u^{(K-1)}) \\ &= u^{(Real)} \end{aligned}$$

5. CONCLUSIONS

Considering the span relations given in the theorem 4, a stratagem of forecast method arrangements is proposed here. If the forecast sequence $\{u^{(K)}\}$ is a non decrease (or increase) sequence, that is $(u^{(K)} - u^{(K-1)})(u^{(K-1)} - u^{(K-2)}) > 0$ and $|u^{(K)} - u^{(K-1)}| < |u^{(K-1)} - u^{(K-2)}|$

Then the forward linear forecast VBF method is used to accelerate the approaching process of the virtual boundary to the real boundary values; otherwise the backward linear forecast VBF method is used to prevent a potential over forecast.

6. REFERENCES

- [1] Barkai, D., Brandt, A. Vectorised multigrid Poisson solvers for the CDC Cyber 205. *Applied Mathematics and Computation*, **13** (1983) 215-227
- [2] Chan, T.F., Schreiber, R. Parallel networks for multi-grid algorithms – architecture and complexity. *SIAM J. Sci. Stat. Comput.*, **6** (1985) 698-711
- [3] Frederickson, P.O., McByran, O.A. Parallel supervonvergent multigrid. Cornell Theory Centre Report CTC87TR12 (1987)
- [4] Naik, V.K., Ta'asan, S. Implementation of multigrid methods for solving Navier-Stokes equations on a multiprocessor system. ICASE Report 87-37 (1987)
- [5] e-Science: Building a Global Grid, DTI e-Science magazine (2002)
- [6] Croft, T.N., Pericleous, K.A., Cross, M. PHYSICA: A multiphysics environment for complex flow processes. Numerical Methods for Laminar and Turbulent Flow (IX/2), C. Taylor et al (eds), Pineridge Press, U.K. (1995)
- [7] Guo, Q.P. et al, Parallel computing using domain decomposition for cyclical temperatures in ceramic/metal composites. SCI 2001 Proceedings, Orlando, USA, ISBN: 980-07-7547-1, 2001.
- [8] Xu, Z. Domain decomposition methods of multi-grid distributed computing. *Journal of Numerical and Computation*, **1** (1996) 1-5
- [9] Proceedings of International Conference on Domain Decomposition Methods for Science and Engineering. Vol 9, 11 and 12, DDM.org.
- [10] Smith, B.F., Bjorstad, P.E. and Gropp, W.D. Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, Cambridge (1996)
- [11] Quarteroni, A. and Valli, A. Domain Decomposition Methods for Partial Differential Equations. Oxford University Press Inc., New York (1999)
- [12] Cai, Xiao-Chuan. A family of overlapping Schwarz algorithms for nonsymmetric and indefinite elliptic problems. In Domain-Based Parallelism and Problem Decomposition Methods in Computational Science and Engineering, ed D.E. Keyes, Y. Saad, D.G. Truhlar. SIAM, Philadelphia (1995)
- [13] Lai, C.-H.. Diakoptics, domain decomposition and parallel computing. *The Computer Journal*, **37** (1994) 840 – 846
- [14] Lai, C.-H., Cuffe, A.M., Pericleous, K.A. A domain decomposition algorithm for viscous/inviscid coupling. *Advances in Engineering Software*, **26** (1996) 151 – 159
- [15] Guo Qingping, Y. Paker, et al., Parallel Multi-grid Algorithm with Virtual Boundary Forecast Domain Decomposition Method for Solving Non-linear Heat Transfer Equation, Lecture Notes in Computer Science, High Performance Computing and Networking, Spreinge Press, May 2000, Vol. 1823, pp568–571.



Guo Qingping is a Full Professor and a head of Parallel Processing Lab, dean of Computer Technology Institute in School of Computer Science and Technology, Wuhan University of Technology. He graduated from Wuhan University in 1968; from Huazhong

University of Science and Technology in 1981 with specialty of wireless technology. He is a holder of K. C. Wong Award of UK Royal Society (1994); was a visiting scholar of City University and University of West Minster (1986~1988), Visiting Professor of the UK Royal Society (1994), Visiting Professor of Queen Mary and Westfield College, London University (1997~2000), Visiting Professor of National University of Singapore (2000), Visiting Professor of University Greenwich (2003). He is one of the DCABES international conference founder, was the chairman of DCABES 2001, co-chair of DCABES 2002, and will be the chairman of DCABES 2004. He has published two books, over 80 Journal papers, edited two DCABES Proceedings. His research interests are in distributed parallel processing, grid computing, network security and e-commerce.

On a Distributed Algorithm for the Solution of Nonlinear Transient Parabolic Problems

C.-H. Lai¹ A. J. Davies²

¹School of Computing and Mathematical Sciences, University of Greenwich
London SE10 9LS, UK

²Department of Mathematics, University of Hertfordshire Herts AL10 9AB, UK
Email: C.H.Lai@gre.ac.uk¹ A.J.Davies@herts.ac.uk²

ABSTRACT

A distributed algorithm is described of solving nonlinear transient parabolic problems. A linearization method based on updating the nonlinear coefficients within an iterative loop is applied to the continuous problem. The distributed algorithm is derived from a Laplace transform of the linearised differential equation followed by a numerical inversion of the solutions of the Laplace transformed equations.

Keywords: Distributed algorithm, nonlinear transient parabolic problems, nonlinear conductivity.

1. INTRODUCTION

Many engineering and applied science problems required the solutions of nonlinear diffusion equations where the nonlinear feature usually comes with the material properties or the conductivity. In the case of unsteady problems a time-marching scheme, usually with time step length restrictions, is required in the solution procedure. These restrictions are usually due to stability criteria of an explicit scheme or the truncation errors of an implicit scheme in approximating the temporal derivative. Computing time of such numerical methods inevitably becomes significant. On the other hand fine grain parallelisation of time stepping becomes difficult and it is almost impossible to achieve a distributed algorithm that is able to yield a de-coupling of the original problem. There are also many problems which require solution details not at each time step of the time marching scheme, but at only a few crucial steps and at the steady state. Therefore effort in finding fine details of the solutions at many intermediate time steps is considered being wasted. Such effort becomes significant in the case of nonlinear problems where a linearization process, which amounts to an inner iteration loop within the temporal marching, is required.

The idea of using a Laplace transform and its numerical inverse in the context of differential equation can be found from the literature for linear elliptic problems [1][4], initially, and for the use with boundary element methods [4][7]. There is also work in extending such method for nonlinear problems. The main purposes of such work have been the removal of the time stepping and combination of such method to boundary element methods as an efficient algorithm. This paper investigates the numerical solutions of a nonlinear time dependent parabolic problems using the concept of a Laplace transform and its numerical inverse in the context of developing a distributed algorithm. First, Laplace transform is applied to a linearisation of the time dependent non-linear

parabolic equation. Second, a distributed algorithm is described of solving the resulting set of linear differential equations in the Laplace space and an approximate inverse Laplace method is described of retrieving the solution of the parabolic equation. Third, numerical experiments are provided to examine the efficiency and accuracy of the distributed algorithm. Finally, discussions and conclusions are included.

2. A MODEL NONLINEAR PARABOLIC PROBLEM

The technique developed in this paper is general enough for higher dimensional problems. Therefore in illustrating the concept only problems with one spatial dimension are presented. The model problem in this paper is the nonlinear time dependent diffusion equation,

$$\frac{\partial u}{\partial t} + g(x, t) = K(u) \frac{\partial^2 u}{\partial x^2} \quad \in (a, b) \times (0, T], \quad (1)$$

subject to suitable boundary conditions, defined at $x = a$ and $x = b$, and initial conditions. Here the thermal conductivity, K , is a given function of u , and g is a given function of x and t .

Temporal integration

The conductivity of the model problem is computed by using an approximation \bar{u} , which is updated in every step of an iterative update process defined in the algorithm below. Each step of the iterative process involves a numerical solution to the equation

$$\frac{\partial u}{\partial t} + g(x, t) = K(\bar{u}) \frac{\partial^2 u}{\partial x^2} \quad \in (a, b) \times (t_i, t_{i+1}]. \quad (2)$$

Here $t_i = i\delta t$, $i = 0, 1, L$, and δt is the time step length of a temporal integration. Let $u^{(n)}(x, t_{i+1})$ and $u^{(n)}(x, t_i)$ be the numerical solution of Eq. (1) at $t = t_{i+1}$ and $t = t_i$ respectively. The iterative update process to obtain the numerical solution $u^{(n)}(x, t_{i+1})$, using $u^{(n)}(x, t_i)$ as the initial approximation to \bar{u} , can be obtained by the algorithm below.

Algorithm 1: Temporal integration from t_i to t_{i+1} .

Initial approximation:- $u^{(0)}(x, t_{i+1}) := u^{(n)}(x, t_i)$;

* This author is supported by EPSRC Grant GR/T10183/01.

$k := 0$;
 Iterate
 $k := k + 1$;
 $\bar{u} := u^{(k-1)}(x, t_{i+1})$; {update \bar{u} }
 Compute $K(\bar{u})$;
 $u^{(k)}(x, t_{i+1}) := \text{Apply a temporal marching step to Eq. (2)}$;
 Until $\|u^{(k)}(x, t_{i+1}) - u^{(k-1)}(x, t_{i+1})\| < \varepsilon$
 $n := k$

Numerical solutions of Eq. (1) at each time step requires to execute Algorithm 1 within an outer iteration loop, with $t_i = i\delta t$, $i = 0, 1, \dots$, where δt is the step length of the temporal integration. Restrictions on δt are inevitable and are due to stability criteria of using an explicit scheme or the truncation errors of using an implicit scheme in approximating the temporal derivative in Eq. (2). This is one of the main disadvantages of a temporal integration technique when δt is small whereas the fine details are not required at all intermediate steps.

The inverse Laplace transform method

Alternatively a Laplace transform can be applied to Eq. (2), now being defined in the time interval $(T_j, T_{j+1}]$,

$$l\left(\frac{\partial u}{\partial t}\right) + l(g(t)) = l\left(K(\bar{u})\frac{\partial^2 u}{\partial x^2}\right) \in (a, b) \times (T_j, T_{j+1}] \quad (3)$$

Here $T_j = j\delta T$, $j = 0, 1, \dots$. Note that the time step counter is denoted as j which is different from i . In other words, $\delta T \neq \delta t$. For a Laplace transform method many intermediate time steps can be removed which leads to the choice $\delta T > \delta t$.

Let

$$l(u) \equiv \int_0^\infty e^{-\lambda\tau} u(x, \tau) d\tau = U(\lambda; x), \quad (4)$$

be the Laplace transform of the function $u(x, t)$. Eq. (3) becomes

$$\lambda U - u(x, T_j) + G = K(\bar{u}) \frac{d^2 U}{dx^2} \in (a, b), \quad (5)$$

subject to suitably Laplace transformed boundary conditions at $x = a$ and $x = b$. Here $G = l(g)$ and

$$\lambda \in \left\{ \lambda_p, p = 1, 2, \dots, m : \lambda_p = p \frac{\ln 2}{T_{j+1} - T_j} \right\}, \quad (6)$$

where m is required to be chosen as an even number [1][2].

Therefore the original problem in Eq. (2) is converted to m independent parametric boundary value problems each as described by Eq. (5), and these problems may be distributed and solved independently in a distributed environment which consists of a number of processors linked by a network. From experience, the value of m is usually a small even number not larger than 10 [1][2]. Numerical experiments as shown in later sections also confirm such experience.

In order to retrieve $u(x, T_{j+1})$, the approximate inverse Laplace transform due to Stehfast [5] given by

$$u(x, T_{j+1}) \approx \frac{\ln 2}{T_{j+1} - T_j} \sum_{p=1}^m w_p U(\lambda_p; x), \quad (6)$$

where

$$w_p = (-1)^{m/2+p} \sum_{k=(1+p)/2}^{\min(p, m/2)} \frac{k^{m/2} (2k)!}{(m/2-k)! k! (k-1)! (p-k)! (2k-p)!}$$

is known as the weighting factor, is used. This approximate inverse Laplace transform is by no means the most accurate one, and it is chosen purely for the purpose of demonstrating the distributiveness of the algorithm.

Note that the coefficient $K(\bar{u})$ of Eq. (5) needs to be updated using the idea discussed in the temporal integration.

Algorithm 2: Inverse Laplace Method from T_j to T_{j+1}

Initial approximation:- $u^{(0)}(x, T_{j+1}) := u^{(n)}(x, T_j)$;

$k := 0$;

Iterate

$k := k + 1$;

$\bar{u} := u^{(k-1)}(x, T_{j+1})$; {update \bar{u} }

Compute $K(\bar{u})$;

Distribute $p := 1$ to $m(j)$

Solve Eq. (5) for $U(\lambda_p; x)$;

End Distribute

Compute $u^{(k)}(x, T_{j+1})$ using inverse Laplace in Eq. (6);

Until $\|u^{(k)}(x, T_{j+1}) - u^{(k-1)}(x, T_{j+1})\| < \varepsilon$

$n := k$

Here $m(j)$ is the number of parametric equations. It is required to solve $m(j)$ parametric systems each described by Eq. (5) resulting to $U(\lambda_p; x)$. In order to solve for $u(x, T)$ as described by Eq. (2), Algorithm 2 needs to be iterated through $T_j = T_1, T_2, \dots, T$, with suitable choices of $m(j)$, in the form of an outer iteration over Algorithm 2. In actual implementation often different values of $m(j)$ are not necessary, and the results shown in this paper use the same number of parametric equations, denoted as \bar{m} , for all values of j .

3. NUMERICAL EXPERIMENTS

Numerical tests were carried out for the case where $g(x, t) = 2x^4 e^{-3t} + x^2 e^{-t}$, $K(u) = u^2$, boundary conditions $u(0, t) = 0$ and $u(1, t) = e^{-t}$, initial condition $u(x, 0) = x^2$, and $T = 1$. The analytic solution of the model problem defined in Eq. (1) is given by $u(x, t) = x^2 e^{-t}$.

Algorithm 1 is used to provide a reference solution for comparison. A backward finite difference for the temporal derivative and a second order finite volume method are used in Eq. (2). The mesh size, $h = 1/2^7$ and the step size, $\delta t = 0.001$, are chosen in the numerical tests. The resulting linear system using the above discretisation is a tri-diagonal system, and the work required to solve such system is defined as one work unit. The reference solution at $t = T$, denoted as u_1 , requires 2171 work units and $\|u(x, T) - u_1\|_2 = 0.000006$.

Algorithm 2 is used to solve the Laplace transformed set of equations and to retrieve an approximate solution, denoted as u_2 , to $u(x, T)$. A second order finite volume method, similar to the one used in Algorithm 1, is applied to discretise Eq. (5), and this resulted to \bar{m} set of linear tri-diagonal equations which can be solved in a distributed environment. The mesh size is also chosen as $h = 1/2^7$. The total sequential work unit is recorded and is divided by \bar{m} in order to obtain the distributed work unit, excluding the overhead of computing the inverse of the Laplace transformed solution, in the distributed computing environment.

Table 1 shows the error $\|u(x, T) - u_2\|_2$ against various \bar{m} . Each column represents the errors using a particular value of δT and the last column shows the errors when $\delta T = \delta t$. There is an optimal accuracy for each column and it takes place at \bar{m} being 4 and 6. By choosing $\delta T = 5\delta t$ the accuracy is not degraded very much as compared to the reference solution obtained by using Algorithm 1. On the other hand the accuracy is only one decimal place less than that obtained by using Algorithm 1 when $\delta T = 25\delta t$. For larger values of δT , say $\delta T = 125\delta t$, the accuracy increases with larger values of \bar{m} , which is a sensible observation for nonlinear problems, because a larger step requires higher number of parametric equations to compensate the loss of accuracy due to truncation errors. These results show that the inverse Laplace method based on Stehfast is accurate enough by taking the choice of $\delta T \leq 25\delta t$. The results also suggest that the value of \bar{m} should not be chosen unnecessarily large.

Table 2 shows the distributed work unit, excluding the work required for the inverse Laplace computation, against various \bar{m} . Again each column represents a particular value of δT . Note that the distributed work units correspond to the optimal accuracy as shown in Table 1 are highlighted. These work units are significantly less than that required in obtaining the reference solution. Note also that the distributed work unit in the last column, i.e. when $\delta T = \delta t$, multiplies the corresponding value of \bar{m} gives the same amount of work

unit as required by using Algorithm 1. This serves to validate the correctness of Algorithm 2.

Table 1: $\|u(x, T) - u_2\|_2$ for various \bar{m} and δT .

$\bar{m} \setminus \delta T$	$125\delta t$	$25\delta t$	$5\delta t$	δt
2	0.016913	0.017783	0.018089	0.018159
4	0.001039	0.000048	0.000348	0.000417
6	0.000504	0.000064	0.000009	0.000002
8	0.000497	0.000094	0.000020	0.000004
10	0.000496	0.000094	0.000019	0.000004
12	0.000495	0.000094	0.000019	0.000004

Table 2: Distributed work unit for various \bar{m} and δT .

$\bar{m} \setminus \delta T$	$125\delta t$	$25\delta t$	$5\delta t$	δt
2	36.5	100	340	1500
4	11.75	38	150	520.5
6	7.67	25.17	100	348.33
8	5.75	18.75	75	261.378
10	4.6	15	60	209.1
12	3.83	12.5	50	174.25

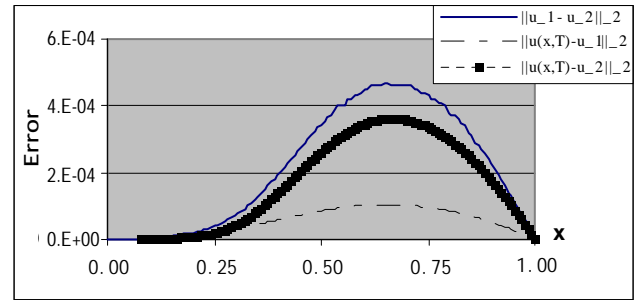


Figure 1: Discrepancy in solutions.

The errors with respect to the analytic solution, $\|u(x, T) - u_1\|_2$ and $\|u(x, T) - u_2\|_2$, and the discrepancy between the reference solution and the inverse Laplace solution, $\|u_1 - u_2\|_2$, are presented in Figure 1. The inverse Laplace solution is correct up to 4 decimal places.

4. CONCLUSIONS

The Laplace transformed equation produces a set of parametric equations which may be solved in a distributed environment. These parametric equations are completely de-coupled from each other during the process of their solutions. The choice of \bar{m} need not be a large value, and experiments show that the most favourable choice is around 4 and 6. Any values greater than 6 are not necessary. The algorithm is suitable to be run on a coarse grained distributed environment with only a few computers linked up by a network. In particular the value of \bar{m} is the number of computers/processors required. The inverse of Laplace is computed as a sequential overhead.

For the present model problem the time step length chosen for the Laplace transform method can be much larger than the time step length chosen for the temporal integration method. Typically δT may be chosen up to $25\delta t$. The sequential work unit recorded for the Laplace transform method when $\delta T = \delta t$ is of the same order of that recorded for the

temporal integration method. Therefore with the distributed environment running the distributed algorithm, one expects the Laplace transform method is still a good method.

Finally the discrepancy in solution between the temporal integration method and the inverse Laplace method is of the order 10^{-4} .

5. REFERENCES

- [1] D. Crann. The Laplace transform: Numerical inversion for computational methods. Technical Report No. 21, University of Hertfordshire, July 1996.
- [2] D. Crann, A.J. Davies, C.-H. Lai, and S.W. Leong. Time domain decomposition for European options in financial modelling. In Proceedings of the 10th International Conference on Domain Decomposition Methods, August 1997, Colorado. J. Mandel, C. Farhat, and X.-C. Cai, editors, Domain Decomposition Methods 10, American Mathematical Society, 1998.
- [3] G.J. Moridis and D.L. Reddell. The Laplace transform finite difference method for simulation of flow through porous media, *Water Resources Research*, **27**, 1873 – 1884, 1991.
- [4] P. Satravaha and S. Zhu. An application of the LTDRM to transient diffusion problems with nonlinear material properties and nonlinear boundary conditions. *Applied Maths and Comput*, **87**, 127 – 160, 1997.
- [5] H. Stehfest. Numerical inversion of Laplace transforms. *Comm ACM*, **13** : 47-49, 1970.
- [6] D.V. Widder. **The Laplace Transform**. Princeton University Press, Princeton, 1946.
- [7] S. Zhu, P. Satravaha and X. Lu. Solving the linear diffusion equations with the dual reciprocity method in Laplace space. *Eng. Anal. Boundary Elements*, **13**, 1 - 10, 1994.

C.-H. Lai is Reader in Scientific Computing, School of Computing and Mathematical Sciences, University of Greenwich, UK.

A. J. Davies is Professor of Mathematics and Head of the Department of Applied Physics and Mathematics, University of Hertfordshire, UK

Parallel and Multilevel Algorithms for Computational Partial Differential Equations

Peter K Jimack*

School of Computing, University of Leeds, Leeds, LS2 9JT, UK

Email: pkj@comp.leeds.ac.uk Tel.: +44 113 343 5464

ABSTRACT

The efficient and reliable solution of partial differential equations (PDEs) plays an essential role in a very large number of applications in business, engineering and science, ranging from the modelling of financial markets through to the prediction of complex fluid flows. This paper presents a discussion of alternative approaches to the fast solution of elliptic and parabolic PDEs based upon the use of parallel, adaptive and multilevel algorithms. Mesh adaptivity is essential to ensure that the solution is approximated to different local resolutions across the domain according to its local properties, whilst the multilevel algorithms ensure that the computational time to solve the resulting finite element equations is proportional to the number of unknowns. Applying these techniques efficiently on parallel computer architectures leads to significant practical problems. Difficulties addressed in this paper include how to handle the coarse grid operations efficiently in parallel and the dynamic load-balancing problem that arises when the finite element mesh is adapted.

Keywords: Partial Differential Equations, Parallel Computing, Multilevel Algorithms, Adaptive Mesh Refinement.

1. INTRODUCTION

In this paper we discuss the efficient numerical solution of elliptic and parabolic partial differential equations (PDEs) based upon the combination of three core ingredients: multilevel solvers, mesh adaptivity and parallel computing. Each of these topics have been actively and broadly studied in their own right in recent years and so it would be unrealistic to attempt to provide a comprehensive introduction to any of them in a short paper such as this. It is clear however that the use of any of these techniques within a computational algorithm has the potential to yield significant enhancements in computational efficiency. Combining any two of these approaches allows the possibility of further efficiency gains at the expense of increased programming complexity, whilst the use of all three has the potential for yet more improvement in performance provided that a number of challenging technical difficulties can be overcome successfully. In this paper we present some of these technical issues and discuss the author's experiences in attempting to address them.

Section 2 below briefly introduces multilevel solvers for elliptic PDEs. In Section 3 the use of mesh adaptivity is discussed, along with its integration with a fast multilevel solution algorithm. Sections 4 and 5 consider the application of these techniques to parallel/distributed computer architectures, and the paper concludes with a brief discussion.

2. MULTILEVEL ALGORITHMS

When solving elliptic PDEs using finite difference (FD) or finite element (FE) discretizations there is a need to produce a computational mesh and then to solve a large sparse system of algebraic equations corresponding to that mesh. It has long been understood that standard direct methods (based around Gaussian elimination) are very inefficient for these equations when the mesh becomes very fine since the solution time grows as $O(N^3)$, where N is the number of degrees of freedom. For structured meshes it is possible to make use of the bandedness of the corresponding algebraic equations in order to improve matters considerably. Similarly, for unstructured meshes it is possible to order the unknowns so as to approximately minimize the fill-in that arises with a direct solver. In neither case however is it possible to solve the algebraic equations at a cost that is even close to $O(N)$ as N becomes large. Standard iterative methods also suffer from a superlinear increase in cost as N grows, due mainly to the condition number of the algebraic equations deteriorating as the mesh spacing becomes small.

One class of solver that is capable of obtaining solutions to the discrete systems of FD or FE equations at a computational cost that approaches $O(N)$ is based upon what are known as multilevel algorithms. The most well-known of such methods are possibly the multigrid algorithms described in sources such as [8,26] for example. The underlying philosophy behind multilevel algorithms is to decompose the problem, and the solution, into components which occupy different levels of a suitable hierarchy. In the case of multigrid these levels are sequences of finer and finer meshes and the solution may be thought of as being built up of contributions from each of these. In the case of a finite element discretization for example, the FE space on each uniformly refined mesh contains each of the spaces corresponding to the coarser meshes. This idea may be generalized to other nested sequences of subspaces too [9,28]. Multilevel additive and multiplicative Schwarz algorithms are also closely related to the classical multigrid approach, as described in [21] for example.

In the following two sections we describe some multilevel algorithms in a little more detail in the context of local mesh refinement and parallel implementation respectively.

3. MESH ADAPTIVITY

For many problems of practical interest the solution contains features that must be resolved on different length scales in different regions of the spatial domain. In such cases it is common to use mesh adaptivity in order to ensure that the solution may be calculated with sufficient accuracy throughout, but without the need to have an equally fine grid everywhere. Such an approach is certainly necessary in order to obtain the best possible computational efficiency for a given numerical simulation. One of the most popular forms of mesh adaptivity is known as h-refinement, [19,22], whereby an initial coarse mesh is locally refined to different levels in different regions,

* URL: <http://www.comp.leeds.ac.uk/pkj>

as dictated by some measure of the error in the representation of the solution throughout the computational domain. This approach may be applied to steady-state problems, using a succession of local refinements, or to time-dependent problems, using a sequence of refinements and derefinements as a transient feature moves within the spatial domain. An example of the latter is illustrated in Figure 1, which shows a mesh at two different times during the adaptive finite element simulation of the rapid solidification of a pure melt using a phase-field model [27] (courtesy of A. Jones). In either case, if the full mesh hierarchy is maintained then it is possible to combine this adaptive technique with the multilevel solution methods suggested in the previous section.

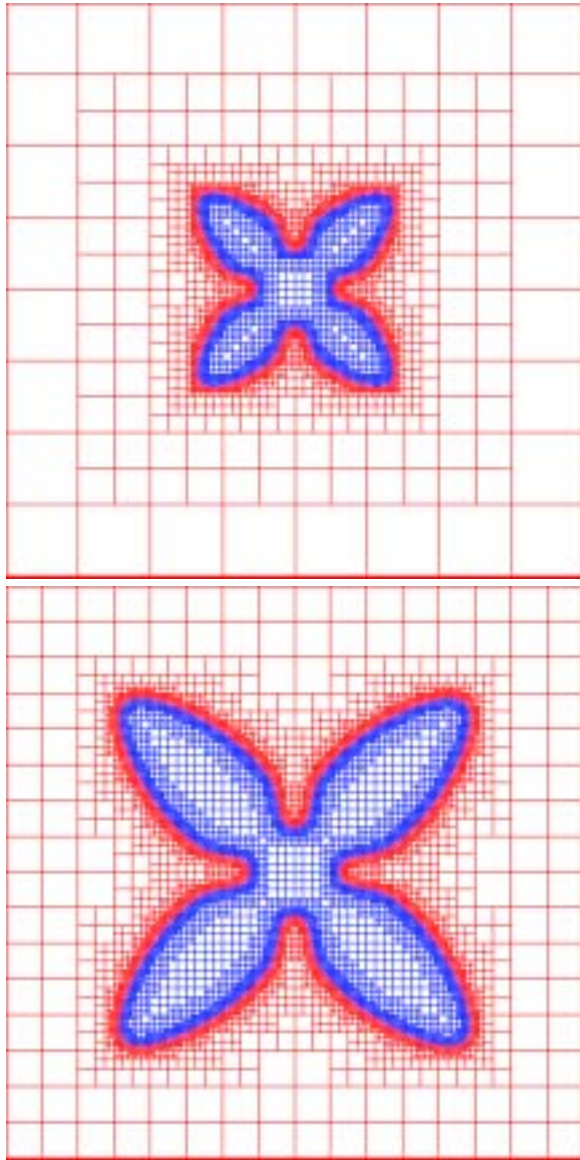


Figure 1 Finite element meshes obtained during the solution of a transient problem using local mesh refinement and derefinement (courtesy of A. Jones).

Work that combines local refinement with the use of multigrid solvers includes the domain decomposition approach of [12] (discussed in more detail in the next section) and the multigrid approaches of [7,10]. In [7] a multilevel adaptive technique

(MLAT) is described in which artificial Dirichlet conditions are imposed at the boundary of all locally refined areas during the solution process. All smoothing at the refined mesh levels is then restricted to within these regions using these temporary boundary conditions. In [10] an alternative approach, known as the fast adaptive composite grid (FAC) method, is described. Here smoothing at each level takes place throughout the entire spatial domain, with suitable modifications made at the interfaces between different levels of mesh refinement. For the remainder of this section a very similar technique to this is described.

In [13] an adaptive multigrid tool is introduced for the solution of elliptic and parabolic PDEs in two dimensions, based upon the use of locally refined mesh hierarchies such as those shown in Figure 1. This is based upon the restriction that no two neighbouring elements of the refined mesh may differ by more than one level and so no elements can possibly have more than one *hanging node* on any edge (these hanging nodes are present only at the interface between different levels of refinement and in the case of the quadrilateral elements appearing in Figure 1 may be identified as the interior nodes that are surrounded by just three elements). In [13] the values of the solution at each hanging node is not a degree of freedom but is prescribed to be the average of the solution values at the two nodes at the end of the edge to which it is the midpoint. In the case of a bilinear FE approximation this ensures that the resulting space only contains continuous functions (i.e. it is conforming). It turns out to be relatively simple to modify the conventional full approximation scheme (FAS) multigrid algorithm in order to accommodate these constraints. The basis of the FAS scheme is shown in Figure 2 for a nonlinear elliptic PDE whose discretizations on two consecutive grids (coarse and fine) are given by $L^C(U^C)=0$ and $L^F(U^F)=0$ respectively. Note that only a two-grid version of the algorithm is provided here for simplicity but the generalization to more mesh levels, in the form of a standard V-cycle for example, is easily achieved by applying the same algorithm to the solution of the coarse grid system.

1. Pre-smooth on the fine grid using nonlinear Gauss-Seidel to obtain U^F .
2. Restrict U^F and $R^F = L^F(U^F)$ to the coarse grid: yielding U^C and R^C .
3. Set the coarse grid correction $C^C = L^C(U^C) - R^C$.
4. Solve on the coarse grid: $L^C(V^C) = C^C$.
5. Set $E^C = V^C - U^C$ (the coarse grid approximation to the error).
6. Interpolate E^C to the fine grid: yielding E^F .
7. Update the fine grid approximation: $U^F = U^F + E^F$.
8. Post-smooth on the fine grid using nonlinear Gauss-Seidel.

Figure 2 The standard FAS multigrid algorithm illustrated with just two levels

The modifications to the standard FAS algorithm for the adaptive finite element scheme used in [13] may be summarized as follows. Firstly, it is necessary that the initial estimate of the fine mesh solution lies in the conforming finite element space. Secondly, the nonlinear smoother that is applied on the fine mesh (steps 1 and 8 in Figure 2) is modified to always project the updated solution into this conforming space. Having obtained this solution and restricted

it and the corresponding residual to the coarse mesh (step 2) the calculation of the coarse grid right-hand side (step 3) and the coarse grid solution (step 4) are both slightly modified to ensure that they are consistent with the conforming approximation. Finally, the modified nonlinear smoother is again applied to the corrected solution (step 8). With these modifications in place it is demonstrated in [13] that a nonlinear elliptic test problem may be solved adaptively such that: a) the adaptive solutions are just as accurate as the solutions on equivalent (much larger) uniform meshes; b) the cost of obtaining an adaptive solution is proportional to the number of degrees of freedom in the mesh; c) the solution may be approximated to a given accuracy at a substantially lower cost using the refinement algorithm. The extension to linear and nonlinear parabolic PDEs using implicit time-stepping is relatively straightforward and allows time steps of an arbitrary length to be selected (free from any stability restrictions for example).

4. PARALLEL IMPLEMENTATION ISSUES

So far we have considered multilevel algorithms and their potential for solving PDE problems in a time that is (close to) proportional to the number of unknowns. The need for mesh adaptivity has also been discussed and it has been shown that it is possible to combine this with a multigrid solver, for example, in a manner that preserves the optimal solution properties of the latter. The final component of an efficient PDE solver that we consider here is that of parallelism. This section therefore focuses on the application of multilevel and adaptive algorithms on distributed memory parallel architectures.

Parallel Multilevel Algorithms

The most straightforward approach to the implementation of a multilevel algorithm in parallel is based upon the use of a geometric partition of the domain. This approach is the most popular since it allows standard sequential algorithms, such as the FAS scheme of Figure 2, to be parallelized without any fundamental alterations. Furthermore, since the assembly of the underlying discrete systems of equations is usually undertaken using a decomposed domain, maintaining this strategy for the parallel solver minimizes the need for data movement within a distributed memory parallel code.

Unfortunately, obtaining good parallel efficiency for the distributed memory implementation of multigrid algorithms such as the FAS scheme is a challenging task. This is primarily due to the fact that a significant amount of computation must be undertaken on relatively coarse meshes which results in undesirably large communication to computation ratios at these levels. Furthermore as the number of processors grows, the coarsest mesh that can be split between the processors in such a way that they each have a non-trivial amount of computation to undertake, also increases in size. Hence the implementer is faced with a choice between making the coarsest mesh finer, which may adversely affect the performance of the multigrid algorithm, or building in processor redundancy (through idle processors or replication of work) at the coarsest levels. The latter approach is generally accepted as being more efficient overall, e.g. [14,16].

Another approach for obtaining optimal, or near-optimal, performance within a parallel solver is through the use of multilevel domain decomposition methods (see, for example,

[21]). These typically fall into two categories: multiplicative and additive. Multiplicative multilevel methods are essentially the same as multigrid methods but derived from a different perspective. Additive multilevel methods however are somewhat different, perhaps the most well known of these being the BPX algorithm of [6]. In this approach the computational work is undertaken independently on all mesh levels before being accumulated (rather than one level at a time for multiplicative methods). This allows more flexibility in the parallelization that just using a geometric decomposition (and, in particular, provides a mechanism for overcoming the coarse mesh issues discussed in the previous paragraph). Examples of the parallel implementation of BPX for uniform meshes can be found in [11,23].

Parallel Adaptive Algorithms

Parallel application of adaptive mesh algorithms, such as those discussed in Section 2, has been considered by a number of authors such as [18,20], for example, which both focus on parallel adaptivity for time-dependent problems in three space dimensions. This is clearly the most general case and therefore illustrates well the various difficulties that arise when implementing adaptive mesh algorithms in parallel.

The first issue that arises has already been discussed in the previous subsection and concerns the geometric partitioning of the mesh data structures. Assuming that local hierarchical refinement takes place, starting with an initial coarse mesh, it is necessary to decide if each element in the mesh hierarchy should always belong to the same processor as its parent (this is known as a vertical partition). If this is the case then the coarsest mesh must have sufficiently many elements for it to be partitioned across the processors with at least one element per processor. Furthermore, as we shall see below, this coarsest mesh must actually contain significantly more elements than there are processors. This assumption has been made in both [18,20].

A further difficulty that arises when undertaking mesh refinement (and coarsening) in parallel is that of maintaining consistent data at the partition boundary. It is very simple for each processor to adapt the interior of its mesh (i.e. those entities with no neighbours owned by another processor) in parallel provided these modifications have no effect on any entities that are not in the interior. In practice this situation occurs only rarely however, and so one of the major overheads in parallel adaptivity turns out to be that associated with communicating alterations to a data structure on one processor to all of the other processors that need to know about them. This task is often eased through the use of halo elements (sometimes referred to as ghost elements or cells) surrounding the sub-mesh that is stored on each processor. These halo elements are copies of the sections of the mesh that are immediate neighbours of the sub-mesh that are actually stored on other processors. Even with their use however these communications can lead to a significant overhead.

As well as the problem of ensuring the consistency of the distributed mesh data structures after parallel adaptivity has occurred, the other major overhead associated with parallel refinement and coarsening of meshes is that of maintaining a good load-balance for the parallel solver. In order to achieve this a parallel dynamic load-balancing algorithm is required which is capable of modifying an existing partition of a mesh so that: a) the total load on each processor is about the same; b) the partition boundary is as short as possible; c) the amount of

data that needs to be migrated is as small as possible. Of course these three criteria are not always mutually consistent and even when the third of them is dropped the resulting problem is known to be NP-hard. Despite this a number of relatively good parallel heuristics do exist (e.g. [15,25]) although it is always necessary to be sure that the costs associated with any migration of data between processors will be more than offset by the improved parallel performance on the new partition.

It should be noted that where a vertical partitioning strategy has been selected only mesh objects associated with the coarsest mesh may be migrated between processors. When such a migration occurs the entire mesh hierarchy beneath this coarse mesh entity must be transferred with it. This clearly means that obtaining a perfect load-balance is unlikely to be possible when some coarse mesh elements have been heavily refined, as in [24] for example. Moreover, a significant amount of communication will be associated with the migration of such data objects. For adaptive algorithms involving both mesh coarsening and refinement this communication may be minimized by first undertaking the mesh coarsening, then migrating the coarse grid data objects in a manner that improves the load balance (based upon knowledge of what refinement is about to take place), and then undertaking the local refinement on the repartitioned mesh. This approach has been implemented with noteworthy success in [18].

5. PARALLEL EXAMPLES

There exists very little software that attempts to combine the three features of multilevel algorithms, mesh adaptivity and parallel implementation in anything approaching their full generality. Noteworthy examples, combining all three aspects for linear elliptic problems, include [4,9]. More recently the work of [4] has been extended to deal with time-dependent problems, as illustrated in [5]. In each of these papers good performance has been reported but in the case of [4,5], for example, the software has required many years of development effort.

Recently, in [1,2], a rather different approach to this problem was proposed for the parallel, adaptive, multilevel solution of elliptic PDEs. This is designed to significantly reduce the development overhead associated with producing such software, provided that an efficient, sequential, adaptive multilevel solver is already available. The approach may be divided into three main steps as follows.

1. A preprocessing step in which the problem is first solved on a single (master) processor using a coarse mesh (which must contain significantly more than p elements, where p is the number of processors). An *a posteriori* error estimate is then used to assign a weight to each coarse mesh element before this mesh is partitioned into p sub-meshes, such that each sub-mesh has an approximately equal total weight.
2. Each processor is allocated a unique subdomain before going on to solve the whole problem independently using the sequential, adaptive algorithm starting with the entire coarse mesh. The only restrictions that are placed on these concurrent solves are that they may only refine inside the subdomain that belongs to the executing processor (or in its immediate vicinity in order to maintain a valid mesh) and that the target number of

elements in the refined grid is approximately equal for each processor.

3. A communication step is now undertaken in which each processor exchanges information with its neighbours, describing its mesh at the interface between the subdomains. Each processor that finds a neighbour with a more highly refined mesh at their interface then adapts its mesh locally in order to ensure that the meshes match at the interface. The union of the refined meshes on each subdomain then defines the partitioned fine mesh over the whole domain. A coupled parallel solve is then undertaken

Figure 3 illustrates this meshing procedure for a simple two-dimensional problem involving two subdomains (above and below the solid bold line in the bottom two meshes). Local refinement has been undertaken in only a small part of the overall domain however the number of elements within each subdomain is approximately equal. Note that in all cases the meshes have been additionally refined in order to ensure that no edge of any element has more than one hanging node (as described in Section 3).

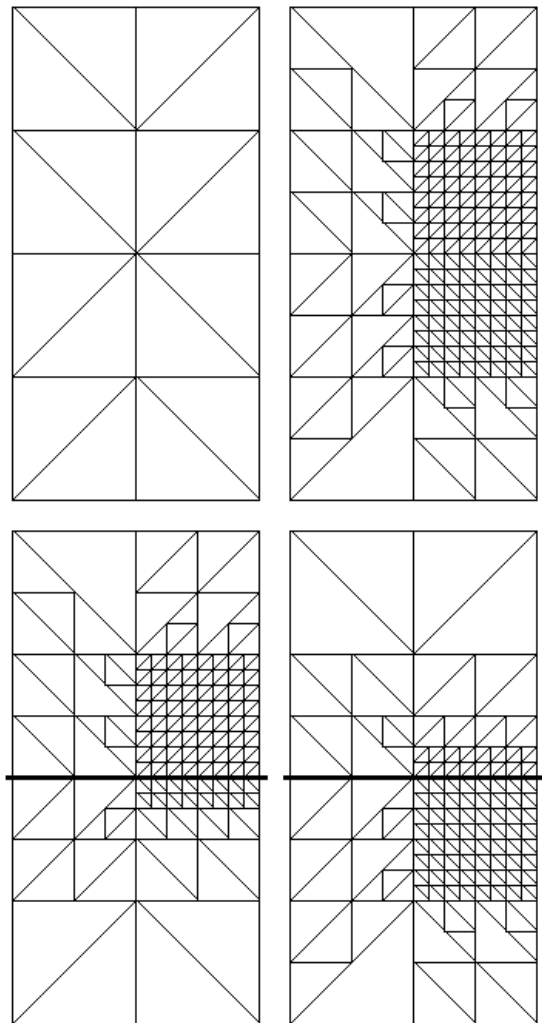


Figure 3 A two-dimensional example illustrating a coarse mesh (top left), a locally refined mesh (top right) and the meshes that are actually created and stored when the algorithm of [1,2] is applied on two processors (bottom left and right).

Having obtained a locally refined mesh on each processor, the final calculation that must be undertaken with this approach (the last part of step three above) is to solve the whole problem in parallel using these meshes. One approach for undertaking this parallel solve is proposed in [1,2] however we now describe an alternative that is introduced in [12]. Both of these approaches are based upon the theoretical results proved in [3] which demonstrate how the solutions on each processor may be combined in a manner that yields the overall solution at an optimal cost, subject to a small number of restrictions on the mesh. These results apply in both two and three dimensions.

The parallel algorithm described in [12] is now summarized. For the simplicity of the explanation it will be assumed that a piecewise linear finite element solution is being calculated using just two processors and that the underlying PDE is linear. For further clarity the reader may find it helpful to consider the two-dimensional case illustrated by the bottom two meshes in Figure 3. Each of the two processors must repeat the iteration shown in Figure 4 until convergence is achieved. As indicated above, in [3] it is proved that for a given class on self-adjoint elliptic PDEs this iteration converges at a rate that is independent of both N and p .

1. Given the latest estimate of the solution find the residual in the subregion owned by the processor.
2. Restrict the residual to the mesh on the other processor that covers the subregion owned by this processor, and send this to the other processor.
3. Receive the residual for the mesh not owned by this processor from the other processor.
4. Given the residual throughout the domain solve for an update of the solution on this processor's mesh using an optimal, multilevel, sequential algorithm.
5. Exchange the update values at the interface and take the average, then update the estimate of the solution.

Figure 4: Outline of the parallel domain decomposition algorithm used in [Refs] for the simplified case of just two processors.

In practice, in [12] the algorithm of Figure 4 is applied as a preconditioner to a GMRES solver rather than as a stationary iteration. Furthermore, it is applied to a class of convection-diffusion equations in three dimensions that is not covered by the underlying theory in [3]. Nevertheless, it proves to be surprisingly robust, as illustrated by the iteration counts shown in Table 1 that are typical of the results in [12]. Furthermore, very creditable parallel performances are recorded, including parallel speed-ups in excess of 12 when using locally refined meshes on 16 processors.

Table 1: The number of iterations of the parallel domain decomposition algorithm required to solve a typical three-dimensional convection-diffusion problem in [12].

Elements / processors	2	4	8	16
2560	5	6	6	6
9728	5	6	6	6
38400	5	6	7	7
153088	6	7	7	7

The iteration counts shown in Table 1 illustrate that the number of iterations of the parallel solver that are required to obtain a converged solution is essentially independent of the level of the finest mesh and the number of subdomains used. Hence, provided the sequential solver used on each processor (at step 4 of the algorithm in Figure 4) has a computational cost of $O(N)$, the total cost of the parallel algorithm will also be approximately proportional to N . This approximation is based upon the assumption that the same coarse grid is always used and that the cost of matching the independently generated meshes at the subdomain boundaries, plus subsequent communication costs, are also $O(N)$. The quality of the load balance that results from the decomposition of the original coarse mesh will not affect this optimal asymptotic performance however it will affect the parallel efficiency which forms part of the constant of proportionality.

6. DISCUSSION

This paper has attempted to introduce and discuss some of the main issues associated with the development of practical software, for the solution of elliptic and parabolic PDEs, that combines the theoretically attractive features of optimal multilevel fast solvers, adaptive mesh refinement and scalable parallel implementation. A variety of multilevel solvers exist for the efficient solution of many elliptic problems and these are also generally applicable for the solution of parabolic equations in combination with (unconditionally stable) implicit time stepping. The parallelization of such solvers is generally undertaken through a domain decomposition approach although more flexibility is possible for additive multilevel algorithms such as BPX: only the domain decomposition methodology is considered here however. When adaptive local refinement is undertaken the different levels in the mesh hierarchy no longer distribute elements uniformly throughout the spatial domain and so the strategy used for the geometric decomposition must become more sophisticated. The use of dynamic load-balancing has been discussed, especially in the context of parabolic problems for which the mesh must be permitted to coarsen in some regions as well as being allowed to refine in others. A suitable strategy is also required for the multigrid smoother used at each level of a locally refined mesh hierarchy: one such algorithm has been outlined based upon the hanging nodes not being treated as true degrees of freedom.

Examples are provided for which all of these considerations have been brought together within a single piece of software however it is noted that this is a highly sophisticated task requiring many years of programming effort. Motivated by this observation an alternative strategy, that is only presently suitable for the solution of elliptic problems, has been described. This strategy assumes that an efficient sequential solver already exists and seeks to integrate it within an outer iteration that is suitable for parallel and distributed computing. The issue of load balance is only addressed in a relatively simple manner however the advantages of this simplicity ensure that a practical parallel implementation may be achieved with relative ease. The two main technical issues that must be addressed are ensuring that the mesh is consistent at subdomain boundaries and accumulating the residual on each processor in the region outside of its own subdomain. In fact the domain decomposition style of algorithm described here has many similarities with an alternative parallel multigrid approach that has been developed in [17]. An interesting

challenge that must now be addressed is this extension of this parallel adaptive solution strategy to time dependent problems.

7. ACKNOWLEDGEMENTS

I would very much like to thank the numerous co-workers with whom I have collaborated on much of the work described in this paper. These include Randy Bank, Martin Berzins, Chris Goodyer, Alison Jones, Sarfraz Nadeem and Mark Walkley.

8. REFERENCES

- [1] R.E. Bank and M. Holst, "A New Paradigm for Parallel Adaptive Meshing Algorithms", *SIAM J. Sci. Comput.*, Vol.22, 2000, pp.1411~1443.
- [2] R.E. Bank and M. Holst, "A New Paradigm for Parallel Adaptive Meshing Algorithms", *SIAM Review*, Vol.45, 2003, pp.291~323.
- [3] R.E. Bank, P.K. Jimack, S.A. Nadeem and S.V. Nepomnyaschikh, "A Weakly Overlapping Domain Decomposition Preconditioner for the Finite Element Solution of Elliptic Partial Differential Equations", *SIAM J. Sci. Comput.*, Vol.23, 2002, pp.1817~1841.
- [4] P. Bastian, S. Lang and K. Eckstein, "Parallel Adaptive Multigrid Methods in Plane Linear Elasticity Problems", *Numer. Linear Algebr.*, Vol.4, 1997, pp.153~176.
- [5] P. Bastian and S. Lang "Couplex Benchmark Computations Obtained with the Software Toolbox UG: Simulation of Transport Around a Nuclear Waste Disposal Site", *Comput. Geosciences*, Vol.8, 2004, pp.125~147.
- [6] J.H. Bramble, J.E. Pasciak and J. Xu, "Parallel Multilevel Preconditioners", *Math. Comp.* Vol.55, 1990, pp.1~22.
- [7] A. Brandt, "Multi-Level Adaptive Solutions to Boundary Value Problems", *Math. Comp.*, Vol.31, 1977, pp.333~390.
- [8] W.L. Briggs, V.E. Henson and S.F. McCormick, *A Multigrid Tutorial* (Second edition, SIAM), 2000.
- [9] M. Griebel and G. Zumbusch, "Parallel Adaptive Subspace Correction Schemes with Applications to Elasticity", *Comput. Methods Appl. Mech. Engrg.*, Vol.184, 2000, pp.303~332.
- [10] L. Hart, S.F. McCormick and A. O'Gallagher, "The Fast Adaptive Composite-Grid Method (FAC): Algorithms for Advanced Computers", *Appl. Math. Comp.*, Vol.19, 1986, pp.103~125.
- [11] B. Heise and M. Jung, "Efficiency, Scalability, and Robustness of Parallel Multilevel Methods for Nonlinear Partial Differential Equations", *SIAM J. Sci. Comp.*, Vol.20, 1998, pp.553~567.
- [12] P.K. Jimack and S.A. Nadeem, "Parallel Application of a Novel Domain Decomposition Preconditioner for the Adaptive Finite-Element Solution of Three-Dimensional Convection-Dominated PDEs", *Concurrency Computat: Pract. Exper.*, Vol.15, 2003, pp.939~956.
- [13] A.C. Jones and P.K. Jimack, "An Adaptive Multigrid Tool for CFD Applications", in *Numerical Methods for Fluid Dynamics VIII*, ed. M.J.Baines et al (Institute of Computational Fluid Dynamics, Oxford), 2004.
- [14] J.E. Jones and S.F. McCormick, "Parallel Multigrid Methods", in *Parallel Numerical Algorithms*, ed. D.E.Keyes, A.Sameh and V.Venkatakrishnan (Kluwer, Dordrecht), 1997.
- [15] G. Karypis and V. Kumar, "Parallel Multilevel k-Way Partitioning Scheme for Irregular Graphs", *SIAM Review*, Vol. 41, 1999, pp.278~300.
- [16] J. Linden, G. Lonsdale, H. Ritzdorf and H. Schuller, "Scalability Aspects of Parallel Multigrid", *Future Gen. Comp. Sys.*, Vol.10, 1994, pp.429~449.
- [17] W. Mitchell, "A Parallel Multigrid Method Using the Full Domain Partition", *Electron. Trans. Numer. Anal.*, Vol.6, 1998, pp.224~233.
- [18] L. Oliker, R.Biswas and R.C. Strawn, "Parallel Implementation of an Adaptive Scheme for 3D Unstructured Grids on the SP2", in *Parallel Algorithms for Irregularly Structured Problems*, LNCS 1117 (Springer-Verlag), 1996.
- [19] N. Provatas, N. Goldenfeld and J. Dantzig, "Adaptive Mesh Refinement Computation of Solidification Microstructures using Dynamic Data Structures", *J. Comput. Phys.*, Vol.148, 1999, pp.265~290.
- [20] P.M. Selwood and M. Berzins, "Parallel Unstructured Tetrahedral Mesh Adaptation: Algorithms, Implementation and Scalability, *Concurrency Computat: Pract. Exper.*, Vol.11, 1999, pp.863~884.
- [21] B. Smith, P. Bjorstad and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations* (Cambridge University Press), 1996.
- [22] W. Speares and M. Berzins, "A 3-D Unstructured Mesh Adaptation Algorithm for Time-Dependent Shock Dominated Problems", *Int. J. Numer. Methods Engrg.*, Vol.25, 1997, pp.81~104.
- [23] M. Thess, "Parallel Multilevel Preconditioners for Thin Smooth Shell Finite Element Analysis, *Numer. Lin. Alg. Applic.*, Vol. 5, 1998, pp.401~440.
- [24] N. Touheed, P. Selwood, P.K. Jimack and M. Berzins, "A Comparison of Some Dynamic Load-Balancing Algorithms for a Parallel Adaptive Flow Solver", *Parallel Computing*, Vol.26, 2000, pp.1535~1554.
- [25] C. Walshaw, M. Cross and M.G. Everett, "Parallel Dynamic Graph Partitioning for Adaptive Unstructured Meshes", *J. Parallel Distrib. Comp.*, Vol.47, 1997, 102~108.
- [26] P. Wesseling, *Introduction to Multigrid Methods* (Wiley), 1992.
- [27] A.A. Wheeler, B.T. Murray and R.J. Schaefer, "Computation of Dendrites using a Phase-Field Model", *Physica D*, Vol.66, 1993, pp.243~262.
- [28] J. Xu, "Iterative Methods by Space Decomposition and Subspace Correction", *SIAM Review*, Vol.34, 1992, pp.581~613.



Peter K Jimack is Professor of Scientific Computing in the School of Computing at The University of Leeds, UK. He graduated from The University of Bristol, UK, with a BSc in Mathematics in July 1986 and a PhD in January 1990, beginning work at Leeds as a Lecturer in Mathematical Software later that year. He has published around 80 papers on topics involving the theory and application of the finite element method, with particular emphasis on adaptive methods and parallel algorithms. His recent publications include work on fast solvers and a wide variety of applications in Computational Fluid Dynamics.

A Parallel Asynchronous Hybrid Method to Accelerate Convergence of a Linear System

Haiwu HE Guy BERGERE Serge PETITON

Laboratoire d'Informatique Fondamentale de Lille, Université de Lille I
Villeneuve d'Ascq Cedex, 59655, France.

E-mail : he@lil.fr, bergereg@lil.fr, petiton@lil.fr Tel : +33328767339

ABSTRACT

We present a parallel hybrid asynchronous method to solve large sparse linear systems by the use of a large parallel machine. This method combines a parallel GMRES(m) algorithm with the Least Squares method that needs some eigenvalues obtained from a parallel Arnoldi's algorithm. All of the algorithms run on the different processors of an IBM SP3 computer simultaneously. This implementation of this hybrid method allows to take advantage of the parallelism available and to accelerate the convergence by decreasing considerably the number of iterations.

Keywords: linear algebra, sparse matrices, iterative method, GMRES, hybrid method, Arnoldi, Least Squares, parallelism.

1. INTRODUCTION

Many scientific applications require the resolution of linear systems of the form $Ax = b$, where A is a $n \times n$ real matrix, b is a real vector, and x the real vector of the solution of the system. Such systems are often implanted by large sparse matrices; this sparse structure is very helpful to solve a large linear system.

The method **GMRES(m)** [7] allows to resolve such linear systems with a very large scale. In addition, it allows computing sparse matrices in compressed formats, without loading zeros in memory, because it preserves sparse structure. It has been implemented on a parallel machine [1], but this method can not always converge very fast. There are some ways to accelerate the convergence of **GMRES**. One of those is to calculate in parallel some eigenvalues by Arnoldi method [2, 6]. As soon as they will be approximated with a sufficient accuracy, eigenvalues are used to perform some iterations of the Least Squares method [5] for getting a better initial vector for the next **GMRES** restarts.

2. THE GMRES(M)/LS-ARNOLDI(K,L) HYBRID PARALLEL METHOD

The hybrid algorithm contains two parts, in the first we apply **GMRES(m)** to solve the linear system. In the second part, we get some eigenvalues estimates. We can take advantage of a restart of GMRES by executing the iterations **LS** when the number of the eigenvalues received is enough. We describe the algorithm as follows

Algorithm: GMRES(m)/LS(k, l)

1. Start: Choose x_0 , m , m' the dimension of Krylov subspaces, k the degree of the Least Squares polynomial, ε the threshold and l the number of successive application of the Least Squares method.
2. Compute x_m , the m^{th} iterate of GMRES starting with x_0 , if $\|b - Ax_m\|_2 < \varepsilon$ Stop else set $x_0 = x_m$, $r_0 = b - Ax_0$.
3. Perform simultaneously m' iterations of Arnoldi process on the other processors starting with r_0 , and compute the eigenvalues of $H_{m'}$.
4. If the number of eigenvalues obtained is sufficient then compute the **Least Square** polynomial P_k on the boundary of H the hull convex enclosing all computed eigenvalues.
5. For $j=1, \dots, l$ do
 Compute $\tilde{x} = x_0 + P_k(A)r_0$, and set $x_0 = \tilde{x}$, $r_0 = b - Ax_0$.
 End do
6. Restart: if $\|r_0\|_2 < \varepsilon$ stop else goto 2.

Suppose that the computed hull convex H contains only the eigenvalues $\lambda_1, \dots, \lambda_s$, so the last residual is given by

$$\tilde{r} = (R_k(A))^l r_0 = \sum_{i=1}^s \rho(R_k(\lambda_i))^l u_i + \sum_{i=s+1}^n \rho(R_k(\lambda_i))^l u_i$$

The first part of the residual is very small because the **LS** method finds R_k minimizing $|R_k(\lambda)|$ with $\lambda \in H$, but not the second part, so the residual will be rich in the eigenvectors associated to the eigenvalues outside the hull convex H .

We can tell that as l increases the first part will be much closed to zero and the second part will be very great, this explains the fact that residual norm increases enormously. However, restarting **GMRES(m)** with an iterate of which the residual norm is enormous.

Because of the reason above, it is better to take the **LS** residual as initial vector of the Arnoldi's method in order to find new eigenvalues outside the hull convex H . This technique is used in the Least Squares Arnoldi method, see [10].

In order to implement this hybrid method, we use the supercomputer “IBM SP3”. A group of processors is in charge of the **GMRES**(m) algorithm, another group takes charge in the parallel Arnoldi algorithm which calculates independently the eigenvalues necessary for the hybridization of the **GMRES**(m) method. The Arnoldi method includes some parts which can be parallelized (Arnoldi’s projection, residuals calculation, restarts). It is realized by the scientific software parallel package “PARPACK” [8] which runs on a group of processors, and a sequential part (the **LS** method and the sorting of eigenvalues) which runs on only a processor.

In the Arnoldi method, when the numbers of the eigenvalues whose residual is under the chosen threshold is sufficient, these eigenvalues will be sent to the processors devoted to the sequential part of hybridization. This computation is sequential because it uses just a small set of data, a parallel distribution of them is unnecessary. This process will calculate “Least Squares” parameters, i.e. parameters of convex polygon containing eigenvalues in the complex plan, parameters of the ellipse of smallest area enclosing this convex polygon, and coefficients of the Least Squares polynomial (see figure 0).

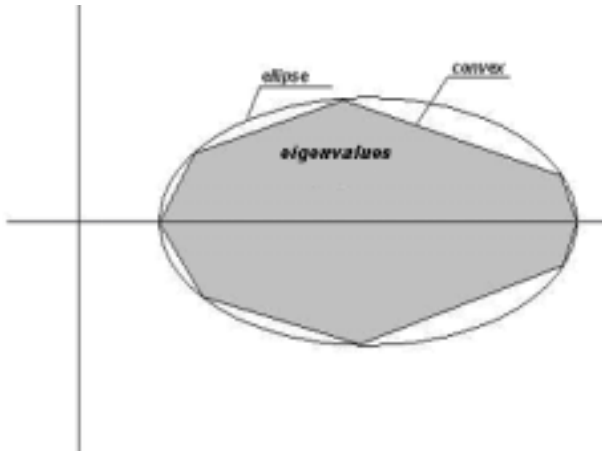


Fig.0 The representation of eigenvalues in the complex plan of a real matrix, the convex and the ellipse including them

These parameters are then sent in order to execute the parallel part of the hybridization. The processes executing **GMRES**(m) algorithm receive these data in an asynchronous manner, at the end of the current iteration. The **GMRES** algorithm is then stopped, and the parallel part of the hybridization is realized, by the Least Squares method, before restarting **GMRES** with the obtained iterate. The residual of this vector is also sent to the processors which run the Arnoldi’s algorithm in order to use it as a new better initial vector for Arnoldi’s method.

The **GMRES** processes show a strong coupling, so the communications are intense and there exists an optimal number of processors. Beyond this number, the time of proceeding increases. In figure 1, we notice 9 is the optimal number for this matrix and on this computer. But when we increase the number of processors to 10 or 11 processors, the time of calculation doesn’t increase a lot because of the mechanism of share memory in just one node of IBM SP3. We can use the available parallelism by executing another algorithm (the calculation of the eigenvalues by the Arnoldi

method). The use of such a hybrid method allows to accelerate the convergence while the improvement of parallelism of the method **GMRES**(m) will be ineffective, even degrade the performance. We remark these two algorithms are asynchronous. The eigenvalues we get allow accelerating the convergence after being proceeded by the method “Least Squares”. The asynchronism allows the postponed reception of the eigenvalues without slowing down the principal calculation of the system resolution.

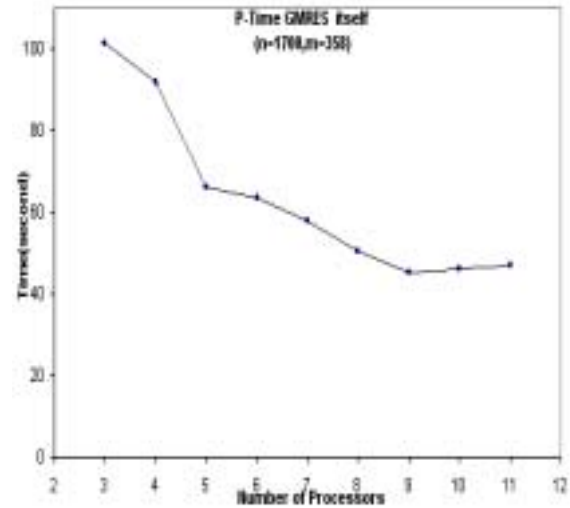


Fig.1 The P-time matrix “utm1700a”

3. IMPLEMENTATION

To have experiments on such a system, we choose the supercomputer system IBM SP3 : one SMP architecture with 5 nodes interconnected by a high debit network. For the 4 original nodes, each node has 16 processors of 375 MHz. For the 5th node updated recently, it has 16 processors of 1.1 GHz. For our experiments, by reason of efficiency, and decreasing the communication of network, we execute our programmes just on one node of the supercomputer.

The most of processors are used to run the algorithm **GMRES**(m) by the way of SPMD model, with an administrative process and p identical calculation processes. The calculation processors read directly their own data and execute the correspondent part of the method **GMRES**(m), communicating with their brother processes.

The processors dedicated to the parallel package “PARPACK” contain the reception of residuals, the projection of Arnoldi and the calculation of eigenvalues. Only a processor calculates the parameters “Least Squares” (the vertexes of convex polygon, the parameters of ellipse (a, c, d) and coefficients of the polynomial “Least Squares” η_i), which will be sent to the processors executing the algorithm **GMRES**(m) later.

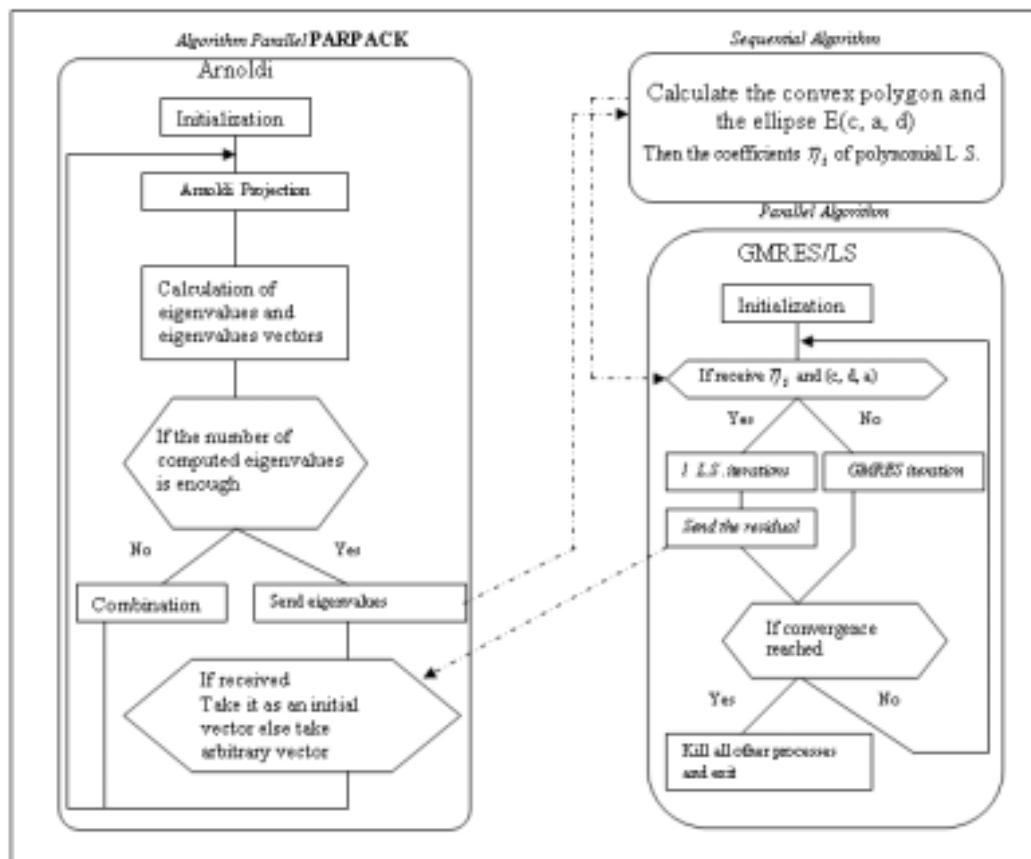
After each iteration, the processors **GMRES**(m) check if the parameters of the method **LS** arrive. In this case, the processes realize the parallel part of the hybridizations, and have a restart of **GMRES**(m). The residual r of this iteration is sent to the administrative process, which forwards it to PARPACK in order to obtain a better restart of the Arnoldi method (see figure 2).

Table.1 K-Iterations for the matrices(NC-Not Convergent)

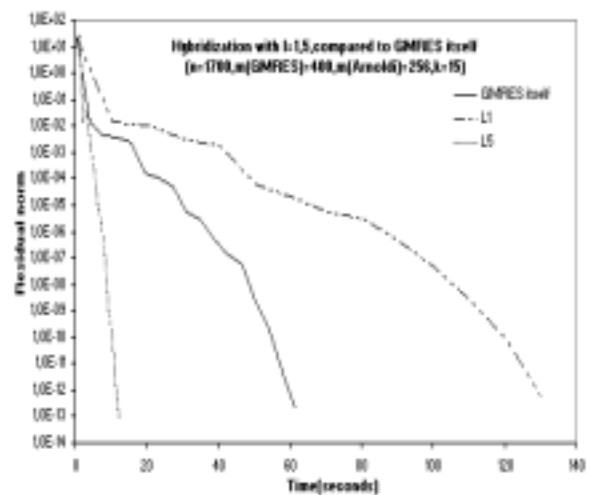
	utm300	utm1700a	af23560
K=10	160	1514	94
K=20	14	849	42
K=30	23	793	NC
K=50	19	875	NC

Table.2 K-Time (second) for the matrices(NC-Not Convergent)

	utm300	utm1700a	af23560
K=10	8.1159	252.0393	318.4566
K=20	0.8325	140.4478	147.4916
K=30	1.5434	134.3323	NC
K=50	1.9122	147.8727	NC

**Fig. 2** General scheme of asynchronous hybrid GMRES/LS-Arnoldi processes

4. NUMERIC RESULTS

**Fig.3** The influence of L of "utm1700a"

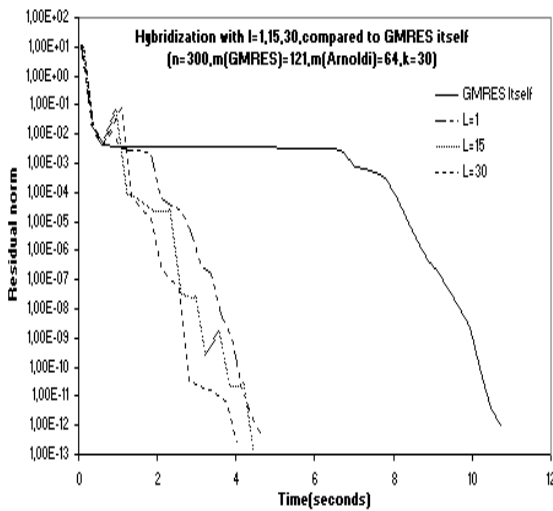


Fig.4 The influence of L of “utm300”

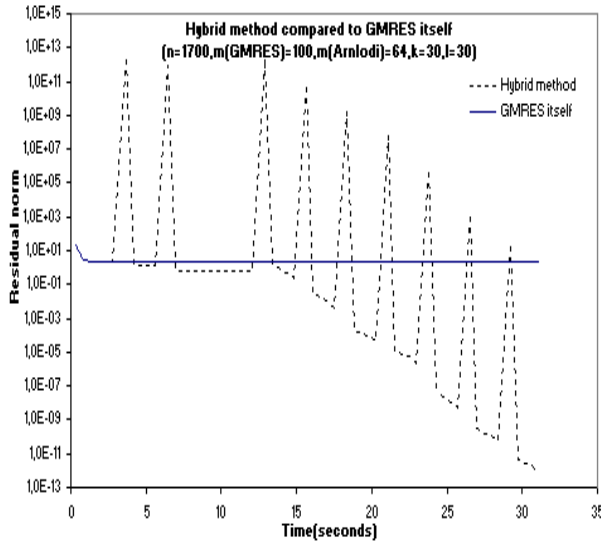


Fig.5 The comparison of hybrid method and GMRES itself of “utm1700a”

The experiments results (see figure 3, 4, 5), tested with the sparse matrices downloaded from the site “MatrixMarket”, showing the acceleration of convergence is unquestionable. We observe that the global computation time of the hybrid parallel method is better than the parallel **GMRES**(m) itself. The speed up can even be spectacular when the convergence of **GMRES** itself is difficult (see figure 5).

When the hybrid computation using **LS** parameters by the **GMRES/LS** processes occurs, we often remark a temporary increase of the residual. However the next decrease of this residual is faster than before the last **LS** iteration, and globally the convergence is accelerated. Thereby, too frequent sending of **LS** parameters damage the efficiency, each **LS** iterate do not have time to have repercussions on many **GMRES** iterations. When the peaks are high and nearby, divergence may even occur.

The non-determinism has a remarkable influence on the performance of the parallel programs. In the mode of time-sharing (see table 3), the performance depends a lot on the charges of the machine. The time of calculation and the number of iterations are almost different for each execution.

We can also remark the same phenomenon happens even for the mode of monopolization (see table 4): Every processes monopolizes a single processor, it don't share the processors with the other processes, but the performance still varies with each execution. Although they don't share the processors at the same time with the others, they have to share the same memory with the other processes that are running on the same node of the machine.

It is possible to improve the efficiency of Least Squares hybridization by calculating a high power l of polynomial. The residual evolution shows high peaks, but the global convergence is faster. However the computation time of parallel part of the hybridization increases in respect to this power. For the small values of l , the time consumed by this computation is less than that one gained with the speed up of convergence. Sometimes the situation is worse, for $l=1$, the hybridization overhead may produce a worse time than with **GMRES**(m) itself (see figure 3). But an excessive increase of l beyond a certain limit offers no additional time, or even a waste of time and we may obtain a divergence.

The **LS** polynomial degree k is also an important parameter, and its value must be sufficient to obtain efficiency with the hybridization. But as the previous parameter, it increases the parallel computation time.

The number of processors (nG) that execute the algorithm **GMRES**(m) is also a key parameter (see figure 6). We can observe that when we put more processors into the calculation after a threshold, that time we used to get the resolution increases contrarily. Because the time of calculation gained by the acceleration with more processors involved in is less than the time consumed by the communication among the processors.

There is a same situation for the number of processors used by the package **PARPACK** (nA). The size of Krylov subspace for Arnoldi method in the package **PARPACK** ($m(Arnoldi)$) has also an obvious influence (see figure 7,8). When we increase this size, we get the eigenvalues more accurate which more helpful to speed up the convergence, but calculating time of eigenvalues increases. So there should exist a balance for choosing a right size.

The experiments show that when increasing l , k , MA (the size of subspace Krylov for the Arnoldi method) time of convergence decreases, then remains at the same value or increases and there are optimal values of l , k , MA . These values appear also for the number of computation iterations. The optimal values of these parameters vary with the matrices and to find them on each case are delicate settings. We can see from the table 1. and table 2. According to the least time and the least numbers of iterations, for the matrix “utm300”, the optimal value of k is 20, for the matrix “utm1700a”, the optimal value of k is 30, for the matrix “af235060”, the optimal value of k is 20. Comparing the table 1. and 2, we notice that sometimes the number of iterations reduces while the time of calculation increases (because of the additional time for **LS** calculation when the parameter k increases).

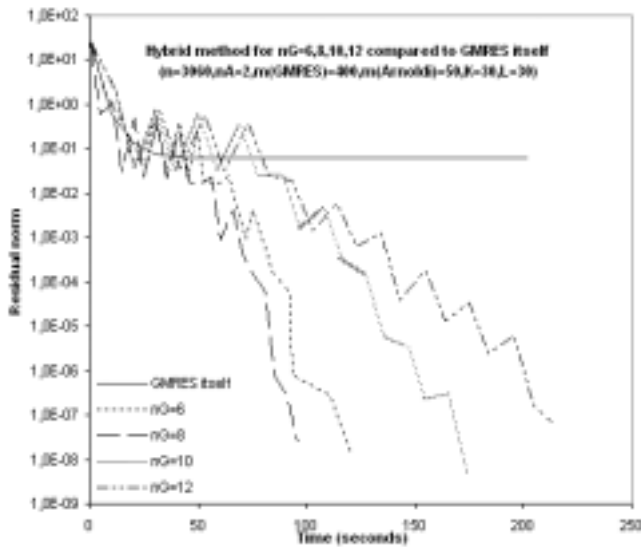


Fig.6 The influence of processors number for GMRES of hybrid method for "utm3060"

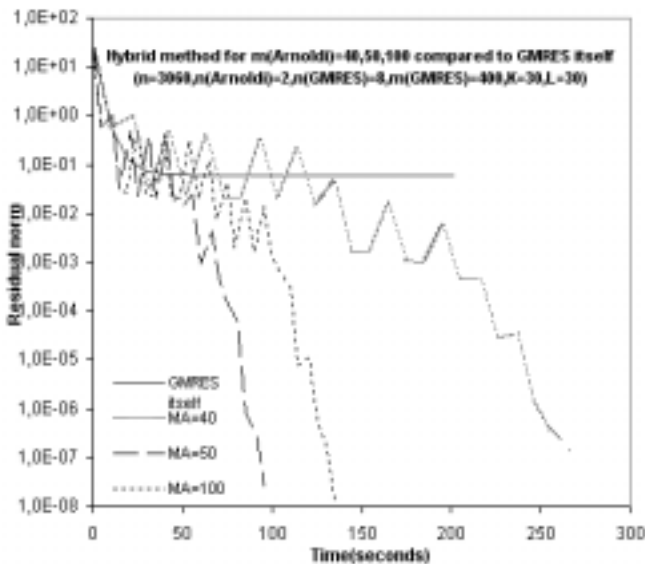


Fig.7 The influence of the size of Krylov subspace in Arnoldi method MA in the hybrid method for "utm3060"

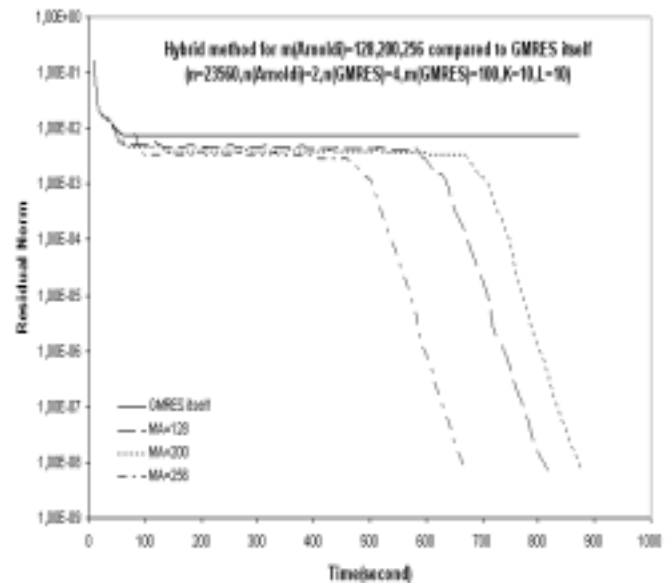


Fig.8 The influence of the size of Krylov subspace in Arnoldi method MA in the hybrid method for "af23560"

Table.3 Non determinism of the acceleration of the convergence (matrix "af23560", Time-sharing)

GMRES(m) Iterations	99	94	99	81
LS Iterations	7	7	8	6
Total Time(second)	903	865	917	706

Table.4 Non determinism of the acceleration of the convergence (matrix "af23560", monopolization)

GMRES(m) Iterations	87	86	111	82
LS Iterations	6	6	8	6
Total Time(second)	301	307	397	273

5. CONCLUSION

The experimental results show the interest of the method even if the evolution of the residual norm presents some peaks. We have obtained very important convergence accelerations, more important that when using just a parallel machine with several nodes, to solve a large scale scientific problem. Thanks to the low amount of communications between its components, our hybrid method takes advantage of available parallelism unusable with the classic method.

The resolution of the linear system by **GMRES(m)** and the calculation of eigenvalues are totally independent. According to the load of the machine, the calculations of eigenvalues are not always realized by the same times of iterations. These calculations are asynchronous with the performance nonreproducible, but in any case this hybridization gives a remarkable acceleration.

The hybrid method is also useful for linear systems with several right-hand sides because the same **LS** polynomial can be used for all right-hand sides.

In near future, we will extend our method to the scientific problems of very large size, the sizes of matrix may reach more than 1×10^6 . In this case, memory available on one node is not sufficient, and we will do more tests on several node of the same computer and on different supercomputers to see the performances.

Another aim is to apply this hybrid method to the distribution computing environment. We will start the tests in mode of peer-to-peer. In this way, the performance may be modest, but we can exploit the many underexploited resources which are much cheaper compared with the supercomputers. Additionally, we are also trying to adapt this hybrid method to resolve the complex problems.

University of Lille) in 2002. Now he works on the domain of supercomputing and distributed calculation.



Guy BERGERE is a doctor of the University of Lille I, France (1999). He works on heterogeneous parallel computing with hybrid numerical methods and control languages on supercomputers or light weight GRID systems

6. REFERENCES

- [1] R.Da Cunha and T.Hopkins, A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations, *Advances in Computational Mathematics*, 2(1994), pp261-277.
- [2] G.Edjlali, N.Emad, S .Petiton, Hybrid methods on network of heterogeneous computers, 14th IMACS World Congress, 1994.
- [3] I.Foster, Task Parallelism and High-Performance Languages, Mathematics and computer Science Division, Argonne National Laboratory, (1995).
- [4] S.Petiton, Parallel subspace method for non-Hermitian eigenproblems on the Connection Machine (CM2), *Applied Numerical Mathematics* 10 (1992), pp19-35.
- [5] Y.Saad, Least Squares Polynomials in the Complex Plane and their Use for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Statist. Comput.*, 7(1987),pp 155-169.
- [6] Y.Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, *Linear Algebra Appl.*, 34(1980), pp.269-295
- [7] Y. Saad, M. H. Schultz, GMRES: A Generalized GMRES Algorithm for Solving Nonsymmetric Linear Systems, *SIAM J. Sci. Statist. Compt.*, 7(1986) 856-869
- [8] R. B. Lehoucq, D. C. Sorensen, C.Yang, ARPACK User's Guide: Solution of Large Scale Eigenvalues Problems with Implicit Restarted Arnoldi Methods, 8 Oct,1997.
- [9] A.Essai, G.Bergère, Serge Petiton, Heterogeneous parallel hybrid GMRES/LS-Arnoldi method. Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999
- [10] Y.Saad, Numerical methods for large eigenvalues problems, Manchester University Press, Manchester (1992).



and Analysis in USTL (Sciences and Technologies

Haiwu HE is a Ph.D student in LIFL (Fundamental Information Laboratory of Lille) of USTL (Sciences and Technologies University of Lille) in France. He graduated for HOHAI University and got Bachelor's degree in 2000 with the major of engineering mechanics. He got his Master's degree in Advanced Instruments

Design and Verification of Parallel Programs*

Wang Jian, Chi Xuebin

Supercomputing Center, Computer Network Information Center, Chinese Academy of Sciences,
Beijing 100080, P.R. China,

Email: {jwang,chi}@sccas.cn Tel: +86-10-62659703

ABSTRACT

Due to the uncertainty and parallelism, the design and verification of parallel programs are far more difficult than the serial ones. This paper introduces them for parallel programs as well as several key problems, modes and verification methods applied in parallel programming.

Keywords: parallel program, verification, design, correctness, specification

1. INTRODUCTION

The design and verification of parallel programs cover both the implementation of computing function and the design of communication mechanism. The exception phenomena, such as dead lock and waiting timelessness, often happen, which trouble the programmers and have no way to be dealt with in the existing compiling systems. It is necessary to study the verification of parallel programs and ensure their correctness. The verification of parallel programs is indispensable in order to make effective use of parallel computing environment, ensure the program to run correctly and safely. This paper presents several methods for verifying the correctness of parallel programs.

The verification is important and necessary in programming. The US space agency struggles with the challenge of creating reliable software. NASA's deep space community is attacking its software crisis via two complementary approaches, one of which is program verification [1]. The verification of parallel programs is far more difficult than that of serial programs. There has not been yet one satisfactory formal system to solve the verification of parallel programs. The theory underlying the verification of the serial programs seems to be fairly well established. In contrast, the corresponding theory for parallel programs has not yet stabilized. A variety of approaches to parallel programs have appeared, ranging from attempts to extend the theory already developed for serial programs to the development of specialized models [9], as well as some informal approaches [10]. [2]

2. DESIGN OF PARALLEL PROGRAM

The design of parallel programs becomes important with the development and application of parallel computer technology. The performance of MPP (Massively Parallel Processor) or cluster has not been to the highest before the support of corresponding software.

Along with the changing of the computer-architecture, during the design of parallel programs, it should be considered how to

harmonize many processors to work well and how to enable the parallel algorithm consistent with the topology of the network architecture. The intrinsic parallelism of the question is applied to the maximum extent and the overhead in the communication among the processors is reduced to the lowest. All the aforementioned are the important questions, which should be taken into account during the design. In short, the design of parallel programs involves many areas, not simply the design of parallel algorithm. Only are many aspects integrated, the effectively parallel programs can be developed.

Generally speaking, it is eventually possible to find a network architecture corresponding to a particular question. The corresponding principle should consist of two points: One is that the data movement should be minimum among the processors or the processors, which communicate with each other, should be as near as possible. The other is the scalability. It is impossible, without any idea of the architecture of the system, to develop a parallel program with high scalability.

At present, there are mainly six modes applied in parallel programming [11]:

- (1) Master-slave
- (2) Single program Multiple Data (SPMD)
- (3) Data Pipelining
- (4) Divide and Conquer
- (5) Speculative Parallelism
- (6) Mixture Model

Both master-slave and SPMD are the best programming models because they are the best in common used models with the global performance.

There are several steps to develop a parallel program: task-division, determination of system topology, process design and system integration. Then an application is built. But there are key problems, which should be considered, during the design of parallel programs. Though the parallel computer systems have obtained rapid development, the development of parallel software is rather lagged and limits the performance of parallel computer. There is lack of parallel programming languages, which is accepted widely. It is difficult to get rid of the influence of serial processing. The overhead among the processors enables the parallel processing technique not worth the candle.

The key problem is how to reduce the overhead of communication, including network interface, delay and receiving-sending overhead, and synchronization. In order to get high performance, it is also necessary to develop the parallel degree. The basic principle is to fine computing granularity and to parallelize them. Due to the high parallelism of the fine computation granularity, the messages are passed at the same time CPU can do other work so as to hide the delay from the message passing.

So far, there are two main approaches to design parallel

*This work was partially supported by National Hitech Program (863)(2002AA104540) and the Special Funds for Major State Basic Research Projects of China (973) (G1999032805)

programs:

First, parallelize the existing serial programs, i.e. serial ones automatically run in parallelism. It mainly studies the operations at the level of loop, including the analysis of data-dependency, data-stream, and the principle of parallel conversion as well as the algorithm supporting the conversion. Generally speaking, automatic parallelism is difficult to mine the maximal parallelism and obtain the optimization. However, it can make full use of the resource of the serial.

Second, start from the question and study the parallel algorithms and parallel programming to solve the question. It mainly focuses on the task-level and how to divide a large task into several (many) independent sub-tasks. Also it should do the correlative analysis about the data and the tasks as well as the parallel implementation. Because it does not depend on the existing serial ones, it can mine the parallelism of the question itself better. Of course, it takes more workload than those of the serial.

3. VERIFICATION OF PARALLEL PROGRAMS

Dijkstra [3] said that programming and verification should proceed hand in hand, meaning that verification is done at the same time that programming is completed. The key point in parallel programming is how to design logically independent processes. Once we have logically independent processes, they can be verified by conventional techniques.

The verification of parallel programs aims at the appearing logical errors in the programs, (mostly, the communicating relation among the processors). The programs are abstracted into one of the verifiable models. The theories and methods of verification are applied to find the errors of the programs, then give the suggestive information or correct them automatically. Until no errors can be found, the program can run well and output the computing results.

3.1 Specifying Programs Correctness [5]

The verification of a program consists in proving that it satisfies a given specification. First, we define one notation:

Definition $Nat = \{0, 1, 2, L\}$.

An informal version of such a specification is, for example, for all input values, $a, b \in Nat$, with $a, b > 0$, the program computes $gcd(a, b)$. (1)

Actually, the specification (1) implicitly assumes that the program terminates. Hence a more precise formulation of the specification is:

for all input values, $a, b \in Nat$, with $a, b > 0$, the program terminates with output value $gcd(a, b)$ (2)

A specification such as (2) is called a specification of 'total correctness'. Now the specification of total correctness (2) is clearly equivalent to the following specification:

for all input values, $a, b \in Nat$, with $a, b > 0$, if the program terminates, then it terminates with the output value

$gcd(a, b)$ (3)
then $\cup (S)(a, b)$ is defined and $\cup (S)(a, b) = gcd(a, b)$.

A specification such as (3) is called a specification of 'partial correctness'.

3.2 Verification methods

3.2.1 Axiomatic Specifications [8]

In an axiomatic method, a specification is written as a list of properties. Formally, Each property is a formula in some logic, and the specification is the conjunction of those formulas. Specification implements specification iff (if and only if) the properties of imply that the properties of are satisfied. In other words, implementation is just logical implication. The obvious language for writing properties is temporal logic [6]. It generates a postcondition for a given unit with its correlative precondition.

3.2.2 Operational Specifications [4]

In an operational approach, a specification consists of an abstract program written in some form of abstract programming language. An obvious advantage of specifying a system as an abstract program is that while few programmers are familiar with temporal logic, they are all familiar with programs. A disadvantage is that a programmer is apt to take it too literally, allowing the specification of what the system is supposed to do bias the implementer towards some particular way of getting the system to do it.

3.2.3 Constructive Methods [8]

It generates one weakest precondition for a chosen unit with its correlative postcondition.

In this approach, a programming language has three sublanguages: one for executable code, one for correctness specifications, and one for proof justifications. A program text consists of code, specifications, and justifications. This approach has a potential weakness, since it requires the programmer to write additional text. The additional effort in writing justifications is partly compensated by the clarity they add. This approach shifts part of the activity of program verification to designing algorithmic notation for proofs and using this notation to express particular proofs. The inventor of this method believes this shift is necessary for theoretical reasons and desirable for methodological reasons. It shows that programs should contain their justifications and that programming languages should be designed accordingly.

3.2.4 Finite State Methods [6]

An important use of state-based methods has been in the automatic verification of finite-state systems. Specifications are written either as abstract programs or temporal-logic formulas, and algorithms are applied to check that one specification implements another.

At present, the verification of parallel programs is most based on the flow:

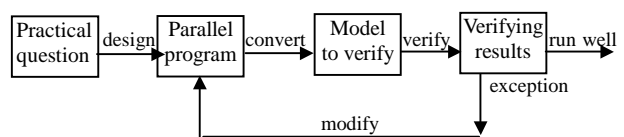


Figure 1

One of the flows during the verification is given here:

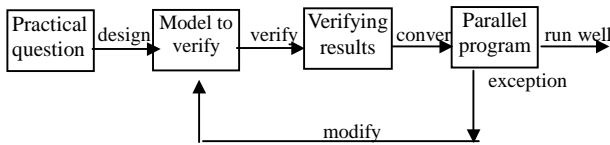


Figure 2

Figure 2 has the theories and methods of verification corresponding to the flow chart of the serial program. The model verification is only processed in the fore end and does not involve the program. When the logical structure is correct, the model is converted into the parallel program. The modification does not involve the particular program. In Figure 1, however, it is necessary to modify the original program when the exception appears. Generally speaking, it is more difficult than to modify the verification model.

4. CONCLUSION

In this paper, the design and verification methods of the parallel program are introduced to the point. In fact, the research in this area is quite abundant. It is one of the most competitive areas in the computer science and technology. Especially, the appearance of network-parallel environment (workstation, cluster) enables the parallel processing to have the diffusive operating environment. Its solution is not only important but also necessary.

5. REFERRNCES

- [1] Patrick Regan, Scott Hamilton, NASA's Mission Reliable, Computer, IEEE, Vol.37, No.1, pp59-68, Jan. 2004.
- [2] Robert M. Keller, Formal verification of parallel programs, Communication of the ACM, Vol.19, No.7, Jul., 1976.
- [3] Dijkstra, E. W. A discipline of programming, Prentice Hall (1976).
- [4] Lamport L. Verification and specification of concurrent programs. - Lect. Notes in Comp. Sci., 1994, v.803, 347-374.
- [5] Jacques Loeckx and Kurt Sieber, Ryan D. Stansifer, The Foundation of Program Verification, Wiley Teubner Series in Computer Science. John Wiley and Sons, 1987. 2nd Edition, pp111-210.
- [6] Leslie Lamport, Verification and specification of concurrent programs, In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, A Decade of Concurrency: Reflections and Perspectives, volume 803 of Lecture Notes in Computer Science, pages 347--374, Berlin, Heidelberg, New York, Jun.1993. Springer-Verlag. Proceedings of a REX School/ Symposium.
- [7] Barry Wilkinson, Michael Allen, Parallel Programming, Techniques and Application Using Networked Workstations and Parallel Computers, Higher Education Press, Beijing, 2002.
- [8] Helmut K. Berg, W. E. Boedert, W. R. Franta, T. G. Moher, Formal Methods of Program Verification and Specification, Prentice-Hall, 1982. pp105-144.
- [9] Keller, R. M. Program schemata and maximal

parallelism, J.ACM 20,3, (July, 1973), 514-537.

- [10] Dijkstra, E.W. Hierachical ordering of sequential processes. Acta Information 1 (1971), 115-138.
- [11] He Yuanqing, Sun Shixin, Fu yan, Analysis of Parallel Programming Model, Journal of UEST of China, Vol.31, No.2, Apr. 2002.

Protein Evolution Based on Complex Networks^{*}

Wang zhongjun^{1,2} Wang nengchao¹

¹School of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan 430071, P.R.China

²Department of Statistics, Wuhan University of Technology
Wuhan 430063, P.R.China

Email: Wangzj3000@vip.sina.com Tel:+86-027-86749365

ABSTRACT

Forming protein structure is a complex evolutionary process. Understanding the evolution of protein structure is useful for prediction of protein structure. The complex network is a effective tool for simulating a complex biological system. Complex networks' evolutionary technology simplifies the evolving process, and helps for analyzing protein evolution and predicting protein structure. This paper discusses the complex protein evolution with complex networks method.

Keywords: complex networks; protein evolution;

1. INTRODUCTION

Proteins are of greatly importance in molecular biology. To understand the structure of protein molecules and how the specific functions are formed is one of the hot topics in protein researches. Recently, there are some rapid progresses in grasping the picture of free energy landscapes of folding, extracting the speciality of structures of protein folds and exploring the possible composition simplicity of proteins. Meanwhile, many problems are still undiscovered. Thus it can be seen, simplified models are the basis of studies on protein structure.

In order to understand protein structure deeply, we apply to the idea of complex system. It has provided us with a first glimpse of the overall evolution of protein structure. So we must study the evolutionary origins of protein structure and evolutionary process. Understanding of the evolutionary origins of protein structures represents a key component of the understanding of molecular evolution as a whole. Protein evolution is a complex system, takes on several basic characters of complex system. Such as hierarchy, coupling, non-linear, evolution, parallel, open and so on.^[1] And complex network is a kind of tool to research complex system.

The task of understanding protein structural evolution has relied on the analysis of structural similarities between proteins. Structural similarity has been defined at varying levels of detail, from the assignment of structures to families and folds in human-annotated databases to the patterns of structural neighbors in quantitative comparisons. Recently, graph theoretic (i.e. complex networks) approaches have been utilized to represent structural similarity at these varying scales, and have been used by many to motivate and implement various models of protein structural evolution.^[2] The complex network is a effective tool for simulating protein evolution system. Complex networks' evolutionary technology simplifies the

evolving process, and helps for analyzing protein evolution and predicting protein structure. This paper discusses the complex protein evolution with complex networks method.

2. COMPLEX NETWORKS

Complex networks describe a wide range of systems in nature and society. They exist in all fields, with a wide range of size, time-scale, interaction mechanism and individual dynamics. Such as the Internet, the World Wide Web, social networks of acquaintance or other connections between individuals, organizational networks and networks of business relations between companies, neural networks, metabolic networks, food webs, distribution networks such as blood vessels or postal delivery routes, networks of citations between papers, and many others.^[3] The analysis of complex networks has attracted great interest in recent years.

2.1 Network Definitions

Systems taking the form of networks are usually called "graphs" in much of the mathematical literature. A network is conveniently modeled as a graph G which consists of a set V of vertices and a set E of edges which we regard as un-ordered pairs of distinct vertices. Hence, we consider only simple undirected graphs in the language of Berge (1985). A path in G is an alternating sequence $(x_0, e_1, x_1, \dots, e_i, x_i)$ of vertices and edges, where the $e_i = \{x_{i-1}, x_i\}$ are the edges connecting subsequent vertices. The length of a path is its number of edges. The set of neighbors of x is denoted by

$$\partial\{x\} = \{y \in V \mid \{x, y\} \in E\}$$

The degree of a vertex x is the number of edges that contain x ; i.e. the number of neighbors of x :

$$\deg(x) = |\{e \in E \mid x \in e\}| = |\{y \in V \mid \{x, y\} \in E\}| = |\partial\{x\}|,$$

where $|A|$ denotes the cardinality (number of elements) of the set A . Equivalently, we may define $\deg(x)$ as the number of edges incident with x .

The distance $d(x, y)$ is the length of the shortest path in G connecting x with y . If a path connecting x and y does not exist we set $d(x, y) = \infty$. Thus, the graph G is connected if and only if $d(x, y)$ is finite for all $x, y \in V$. We remark that our approach can trivially be extended to weighted graphs. So, simply define $\deg(x)$ as the sum of the weight of the edges that contain x and define the length of a path as the sum of the weights of its edges.^[4]

2.2 Network Topological Properties

In order to understand complex networks, we first study network topological properties. Network topological properties include degree distribution, clustering, shortest path, betweenness, and spectrum.

^{*}Research supported by the Natural Science Foundation of China (NSFC Grant No.70371063).

Degree is the number of edges that a node has, and corresponds to the local centrality in social network analysis, and a measure how important is a node with respect to its nearest neighbours. The spread in the node degree is characterized by a distribution function $P(k)$, which gives the probability that a randomly selected node has exactly k edges. In a random graph of the type studied by Erdos and Renyi, each edge is present or absent with equal probability, and hence the degree distribution is binomial or Poisson in the limit of large graph size. Real-world networks are mostly found to be very unlike the random graph in their degree distributions. The degrees distribution has a power-law tail, $P(k) \sim k^{-\gamma}$.

Clustering is a common property of social networks, represents that cliques form. The inherent tendency to cluster is quantified by the clustering coefficient. Clustering coefficient of a node is Clustering coefficient of the whole network is the average of all individual C_i 's

In a random graph, since the edges are distributed randomly,

$$C_i = \frac{E_i}{k_i(k_i-1)/2}$$

$$C = \frac{1}{N} \sum_{i=1}^N C_i$$

the clustering coefficient is $C = p$. However, in most, if not all, real networks the clustering coefficient is typically much larger than it is in a comparable random network.

The shortest (i.e. geodesic) path length is the number of edges that make up the path between two points. It can measure global centrality which points that are "close" to many other points in the network. Global centrality defined as the sum of minimum distances to any other point in the networks.

Between measures the "intermediary" role in the network. It is a set of matrices, B_{ij}^k of each vertex is ratio of shortest paths between i and j that go through k .

There can be more than one geodesic between i and j . The node with the maximum betweenness plays a central role.

$$\rho(\lambda) = \frac{1}{N} \sum_{j=1}^N \delta(\lambda - \lambda_j)$$

Spectrum of the adjacency matrix is a set of eigenvalues of the adjacency matrix. Spectral density (density of eigenvalues) is a symmetric and real network corresponds eigenvalues are real and the largest is not degenerate. The largest eigenvalue shows the density of links. The second largest is related to the conductance of the graph as a set of resistances.

2.3 Three main models

According to their topological properties, complex networks can be divided three main classes of modeling. First, random graphs, which are variants of the Erdos-Renyi model, are still widely used in many fields and serve as a benchmark for many modeling and empirical studies. Second, motivated by clustering, a class of models, collectively called small-world models, has been proposed. These models interpolate between the highly clustered regular lattices and random graphs. Finally, the discovery of the power-law degree distribution has led to the construction of various scale-free models that, by focusing on the network dynamics, aim to offer a universal theory of network evolution. The three main models correspond to three different networks, i.e. random network, small-world network, and scale-free network.

2.4 Property of Evolving Networks

Networks of many interacting units occur in diverse areas as, for example, gene regulation, neural networks, food webs in

$$0 \leq B_{ij}^k \leq 1$$

ecology, species relationships in biological evolution, economic interactions, and the organization of the Internet. So the recent advances in statistical modeling of complex networks have brought the community's attention towards large networks whose topology evolves in time. Network models of evolving topology have been studied with respect to critical properties, such as growth, preferential attachment (k) and so on.

- **Growth:** Starting with a small number (m_0) of nodes, at every time step, we add a new node with $m(<m_0)$ edges that link the new node to m different nodes already present in the system.

- **Preferential attachment:** When choosing the nodes to which the new node connects, we assume that the probability Π that a new node will be connected to node i depends on the degree k_i of node i , such that

$$\Pi(k_i) = \frac{k_i}{\sum_j k_j}$$

After t time steps this procedure results in a network with $N = t + m_0$ nodes and mt edges.^[3]

In most real world networks, they describe open systems that grow by the continuous addition of new nodes. Starting from a small nucleus of nodes, the number of nodes increases throughout the lifetime of the network by the subsequent addition of new nodes. Most real networks also exhibit *preferential attachment*, such that the likelihood of connecting to a node depends on the node's degree, i.e. rich is richer. It suits for biological evolutionary rule that strong is stronger.

3. PROTEIN EVOLUTION NETWORKS

Proteins are very efficient, highly optimized molecular machines that are products of an extremely long evolution-driven design process. We can learn about how proteins evolve from the analysis of structural similarities between proteins, and study similarities in primary sequence or other characters. Structural similarity has been defined at varying levels of detail, from the assignment of structures to families and folds in human-annotated databases (Karev et al., 2002; Koonin et al., 2002; Qian et al., 2001) to the patterns of structural neighbors in quantitative comparisons (Dokholyan et al., 2002). Recently, graph theoretic approaches have been utilized to represent structural similarity at these varying scales, and have been used by many to motivate and implement various models of protein structural evolution.

Many proteins consist of a number of recognizable domains that appear often in many different proteins. A graph G can be constructed that has the domains as its vertices and edges between them whenever two domains co-occur in a protein. Essentially, they give a tentative insight into the structure of the proteome since they were found to exhibit scale-free behavior. Thus, domains which prove to be highly connected since they frequently occur in multidomain proteins shape the backbone of the proteome of the underlying organism.^[4] One particular application called the protein domain universe graph or PDUG (Dokholyan et al., 2002), revealed that the distribution of the

number of structural neighbors k per domain follows a power law $p(k) \sim k^{-\gamma}$ and represents a scale-free network (Albert and Barabasi, 2002; Barabasi and Albert, 1999).^[2]

In the same way, protein-protein interaction network is a kind of protein evolutionary network. In the protein-protein interaction networks, the vertices are proteins and they are connected if it has been experimentally demonstrated that they bind together, and a set of directed or undirected edges, representing the interactions between them (either direct physical interactions or functional associations). A study of these physical interactions shows that the degree distribution of the physical protein interaction map for yeast follows a power law with an exponential cutoff

$$P(k) \sim (k + k_0)^{-\gamma} e^{-(k + k_0)/kc}$$

with $k_0 = 1$, $kc = 20$, and $\gamma = 2.4$ (Jeong, Mason, *et al.*, 2001).

During folding a protein takes up consecutive conformations. Representing with a vertex each distinct state, two conformations are linked if they can be obtained from each other by an elementary move. Scala, Amaral, and Barthelemy (2001) studied the network formed by the conformations of a two-dimensional lattice polymer, finding that it has small-world properties. In particular, evolving networks can be modeled through growing graphs to simulate protein evolution, i.e., graphs to which continuously new vertices and new links (edges) are added.^[7]

4. CONCLUSION

A number of biological systems can be usefully represented as complex networks. Many biological networks, such as metabolic interaction and protein-protein interaction networks have been shown to be scale free, with different vertices having widely different connectivity. Here, the complex networks' evolutionary technology applies to the protein evolving process, and helps for analyzing protein evolution and predicting and restructuring protein structure.

5. REFERENCES

- [1] Zhongjun Wang, Nengchao Wang, and Xingqin Cao, Research on Complex Biological Systems with Automata Network Based on the Evolution Technology of Bisection, DCABES2002, P303-305.
- [2] Eric J., Nikolay V. Dokholyan, and Eugene I. Shakhnovich, Protein Evolution within a Structural Space, Biophysical journal, Volume 85, November 2003, P2962-2972.
- [3] Reka Albert and Albert-Laszlo Barabasi, Statistics Mechanics of Complex Networks, Reviews of modern Physics, volume 74, January 2002, P47-97.
- [4] Stefan Wuchty, Peter F. Stadler, Centers of Complex Networks, journal of Theoretical Biology 223(2003), P45-53.
- [5] Eric Alm and Adam P Arkin, Biological Networks, current Opinion in Structural Biology 2003,13,P193-202.
- [6] Söding J, Lupas AN. (2003). More than the sum of their parts: On the evolution of proteins from peptides. Bioessays 25:837-846.
- [7] J.Jost and M.P.Joy, Evolving networks with distance preferences, PHYSICAL REVIEW, E66, 036126 (2002)
- [8] Yan Li, Scale-free Networks in Biological Complexity,

Phycs498BIO, Assignment 4

- [9] Eli Eisenberg and Erez Y. Levanon, Preferential Attachment in the Protein Network Evolution, Physical Review Letters, volume 91, number 13, P138701-4



Wang Zhongjun is an Associate Professor in the Department of Statistics, School of Science, Wuhan University of Technology. She graduated from Beijing normal University, in 1987, and from Wuhan University of Technology in 1997 and acquire M.S. degree. Now she is Ph.D. candidate of department of computer science, Huazhong University of Science and technology, and majors on theory of computer. She has been a visiting scholar of supercomputing center of computer network information center Chinese Academy of Sciences about two month in 2002, is a study visiting scholar to Budapest University of Technology and Economics in 2003-2004. She has published over 15 Journal papers, and presides several projects about Nation Natural Science Foundation of China, University Natural Science Foundation, Teaching Research and so on. Her research interests are in high Performance Computation, Complex System and bioinformatics.

Algorithm of Decomposition and Reconstruction with Orthogonal Multiwavelet Packets with Random Scale*

Leng Jinsong Huang Tingzhu

School of Applied Math., University of Electronic Science and Technology of China
Chengdu, 610054, P. R. China

E-mail: l-js2004@tom.com Tel.: (028)8904089

ABSTRACT

This paper is based on the construction of orthogonal multiwavelet packets with random scale. They are more flexible in their application. The space $L^2(R)$ can be decomposed by using these multiwavelet packets. Finally the algorithm of decomposition and reconstruction with these orthogonal multiwavelet packets is the main result of this paper.

Keywords: orthogonality, multiwavelet, multiwavelet packets, scale, scaling vector, matrix series with random scale.

1. INTRODUCTION

The symmetry of wavelet basis is important in signal compression. But the scalar orthogonal wavelet basis with compactly support has not symmetry. Multiwavelet initiated by Goodman et al.[1] overcome that drawback. Since then multiwavelet have received considerable attention from the wavelet research communities both in theory and in applications. For consideration of multi-pass filter and more flexible wavelets than two-scale wavelets, Geronimo et al.[2] introduced the theory of multiwavelet with scale=a and get wonderful results. Marassovich [3] composed and studied biorthogonal wavelet.

Wavelet packets were introduced by Coifman et al.[4,5] to improve the poor frequency localization of wavelet basis and thereby provide a more efficient decomposition of signals containing both transient and stationary components. The advantages of wavelet packets and their promising features in application have attracted a great deal of interest and effort in recent years to extensively study them. To only mention a few references here, see [6-12]. Coifman et al.[5] introduced biorthogonal wavelet packets using splines. Yang Shou-zhi et al.[10] provided a method of construction of orthogonal multiwavelet packets. Leng Jin-song *et al.*[11, 12] discussed the biorthogonal case and studied their properties.

In this paper, we introduce the recipe for construction of orthogonal multiwavelet packets with random scale at first. More multiwavelet packets can be constructed from the same orthogonal multiwavelet by using the recipe. We can decompose the space $L^2(R)$ with the multiwavelet packets. Specially, the important algorithm of decomposition and reconstruction with the orthogonal multiwavelet packets with scale=a is given finally. We adopt symbols same as paper [11].

2. ORTHOGONAL MULTIWAVELET PACKETS WITH RANDOM SCALE

Let's begin with some basal knowledge. Then we introduce the construction of orthogonal multiwavelet packets with scale=a.

A vector function $\Phi = \{\varphi_1, \varphi_2, \Lambda, \varphi_t\}$ is said to be orthogonal scaling vector if $\langle \Phi(\cdot), \Phi(\cdot - n) \rangle = \delta_{0,n} I_t$ for $n \in Z$ and it can generate a multiresolution analysis $\{V_j\}_{j \in Z}$. A vector function $\Psi = \{\psi_1, \psi_2, \Lambda, \psi_{(a-1)t}\}$ is said to be orthogonal multiwavelets if $\langle \Phi(\cdot), \Psi(\cdot - n) \rangle = 0$ and $\langle \Psi(\cdot), \Psi(\cdot - n) \rangle = \delta_{0,n} I_{(a-1)t}$ for $n \in Z$. And there exist some sequences of matrices $\{P_k\}$ and $\{Q_k\}$ such that

$$\Phi = \sum_{k \in Z} P_k \Phi(ax - k), \quad \Psi = \sum_{k \in Z} Q_k \Psi(ax - k)$$

and their Fourier transform yield

$$\hat{\Phi}(\omega) = P(Z) \hat{\Phi}\left(\frac{\omega}{a}\right), \quad \hat{\Psi}(\omega) = Q(Z) \hat{\Psi}\left(\frac{\omega}{a}\right)$$

where $P(z) = \frac{1}{a} \sum_{k \in Z} P_k z^k$, $Q(z) = \frac{1}{a} \sum_{k \in Z} Q_k z^k$, $z = e^{-i\omega/a}$ are called matrix symbols of $\{P_k\}$ and $\{Q_k\}$ respectively.

If we define

$$\varphi_{l,j,k} = a^{j/2} \varphi_l(a^j x - k), \quad \psi_{l,j,k} = a^{j/2} \psi_l(a^j x - k)$$

$$V_j = \text{close}_{L^2(R)} \langle \varphi_{l,j,k} : 1 \leq l \leq t, k \in Z \rangle, \quad W_j = \text{close}_{L^2(R)} \langle \psi_{l,j,k} : 1 \leq l \leq (a-1)t, k \in Z \rangle$$

then

$$V_{j+1} = V_j \oplus W_j$$

So $\{\psi_{l,j,k} : 1 \leq l \leq (a-1)t, j, k \in Z\}$ is orthogonal wavelet basis with multiplicity t and scale=a of $L^2(R)$.

For construction of orthogonal multiwavelet packets, we divide Ψ into $a-1$ function vectors with dimension t arbitrarily as follow

$$\Psi_i = (\psi_{i1}, \psi_{i2}, \Lambda, \psi_{it}), \quad i = 1, 2, \Lambda, a-1$$

We define $W_j^{(i)}$ as follow

$$W_j^{(i)} = \text{close}_{L^2(R)} \langle \psi_{im,j,l} : 1 \leq m \leq t, l \in Z \rangle, \quad i = 1, 2, \Lambda, a-1$$

Then we have

$$W_j = \bigoplus_{i=1}^{a-1} W_j^{(i)}$$

And we divided Q_k into $a-1$ $t \times t$ matrices according to the dividing of Ψ_i as follow

* This work is supported by NSFC (60372012).

$$Q_k = (Q_k^{(1)T}, Q_k^{(2)T}, \Lambda, Q_k^{(a-1)T})^T$$

Suppose

$$P_k^{(0)} = P_k, \quad P_k^{(i)} = Q_k^{(i)}, \quad i=1,2,\Lambda, a-1, \quad k \in Z,$$

$$\Psi_0(x) = \Phi(x)$$

then we can get the following definition.

DEFINITION. The vector collection $\{\Psi_{al+i} : l=0,1,\Lambda, a-1\}$ are called orthogonal multiwavelet packets with scale a associated with Φ , where

$$\Psi_{al+i} = \sum_{k \in Z} P_k^{(i)} \Psi_i(ax-k) \quad (1)$$

The matrices $P^{(i)}(z)$ are called matrix symbols of $\{P_k^{(i)}\}$, where

$$P^{(i)}(z) = \frac{1}{a} \sum_{k \in Z} P_k^{(i)} z^k, \quad i=0,1,\Lambda, a-1, \quad z = e^{-i\omega/a} \quad (2)$$

Any $n \in Z$ can be expressed by

$$n = \sum_{j=1}^{\infty} \varepsilon_j a^{j-1}, \quad \varepsilon_j \in \{0,1,2,\Lambda, a-1\}$$

We can prove the frequency field can be divided into tinier ones with wavelet packets of this paper and it can enlarge the appearance of high frequency [11].

3. ALGORITHM OF DECOMPOSITION AND RECONSTRUCTION

In this section, we first decompose the space $L^2(R)$. Then we give algorithm of decomposition and reconstruction. It can decompose a function into tinier levels and reconstruction it by using wavelet packets of this paper. It is important in application.

We can get the following theorem 1 ~ 3 with the same methods as paper [11].

THEOREM 1. Suppose $P^{(i)}(z)$, $i=0,1,\Lambda, a-1$, $z = e^{-i\omega/a}$ is dined by Eq. (2), then

$$\sum_{l=0}^{a-1} P^{(l)}(b_l z) P^{(j)*}(b_l z) = \delta_{i,j} I_t, \quad i, j=0,1,\Lambda, a-1 \quad (3)$$

where $b_l, l=0,1,\Lambda, a-1$ are a roots of equation $z^a - 1 = 0$.

THEOREM 2. Let $\{\Psi_n\}$ are orthogonal multiwavelet packets defined by Eq. (1), then for all $n \in Z_+$

$$\langle \Psi_n(\cdot - k), \Psi_n(\cdot) \rangle = \delta_k I_t, \quad k \in Z$$

THEOREM 3. Let $\{\Psi_n\}$ are orthogonal multiwavelet packets defined by Eq. (1), then

$$\langle \Psi_{al}(\cdot - k), \Psi_{al+i}(\cdot) \rangle = 0, \quad i=1,2,\Lambda, a-1, \quad k \in Z$$

We define X_j^l as follow

$$X_j^l := \text{close}_{L^2(R)} \langle a^{j/2} \Psi_l(a^j x - m), m \in Z \rangle, \quad j \in Z, \quad l \in Z_+$$

Using the above three theorems we can get the following

theorem 4 ~ 5 easily.

THEOREM 4. For all $l \in Z_+$

$$X_j^l = \bigoplus_{i=0}^{a-1} X_j^{al+i}$$

THEOREM 5. For all $j \in Z_+$

$$W_j^{(i)} = \bigoplus_{l=0}^{a^k-1} X_{j-k}^{ia^k+l} = \bigoplus_{l=0}^{a^j-1} X_0^{ia^j+l}, \quad i=1,2,\Lambda, a-1, \quad k \in Z_+$$

More we have

$$L^2(R) = \Lambda \oplus (W_{-1}^{(1)} \oplus W_{-1}^{(2)} \oplus \Lambda \oplus W_{-1}^{(a-1)}) \oplus X_0^l$$

So $\{\Psi_{i,j,l}, \Psi_n(x-l) : 1 \leq i \leq (a-1)t, j \in Z_-, n \in Z_+, l \in Z\}$ are orthogonal basis of $L^2(R)$.

THEOREM 6. If $\{\Psi_n\}$ are orthogonal wavelet packets given by Eq. (1), then for all $k \in Z$

$$\Psi_n(ax-k) = \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{l \in Z} P_{k-al}^{(j)*} \Psi_{an+j}(x-l) \quad (4)$$

Proof. Using Eq.(3), we have

$$\begin{aligned} & \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{l \in Z} P_{k-al}^{(j)*} \Psi_{an+j}(x-l) \\ &= \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{l \in Z} P_{k-al}^{(j)*} \sum_{m \in Z} P_m^{(j)} \Psi_n(ax-al-m) \\ &= \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{l \in Z} \sum_{m \in Z} P_{k-al}^{(j)*} P_m^{(j)} \Psi_n(ax-al-m) \\ &= \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{l \in Z} \sum_{r \in Z} P_{k-al}^{(j)*} P_{r-al}^{(j)} \Psi_n(ax-r) \\ &= \frac{1}{a^2} \sum_{j=0}^{a-1} \sum_{r \in Z} \Psi_n(ax-r) \sum_{l \in Z} P_{k-al}^{(j)*} P_{r-al}^{(j)} \\ &= \Psi_n(ax-k) \end{aligned}$$

So we get Eq. (4).

Given level N , we consider

$$f \approx f_N := \sum_{j \in Z} C_j \Psi_0(z^N x - j) \in V_N$$

where $\{C_j\}$ are constant vector sequence with dimension t .

But

$$V_N = W_{N-1} + V_{N-1} = \Lambda + W_{N-1} + W_{N-2} + \Lambda + W_{N-M} + V_{N-M}$$

So

$$f_N = g_{N-1} + g_{N-2} + \Lambda + g_{N-M} + f_{N-M}$$

where $f_{N-M} \in V_{N-M}$ and $g_j \in W_j, j = N-M, \Lambda, N-1$.

According to Theorem 2 ~ 5, we can decompose $g_j \in W_j, j = N-M, \Lambda, N-1$ deeply.

Let

$$f_j(x) = \sum_{k \in Z} C_k^j \Psi_0(a^j x - k), \quad g_j(x) = \sum_{i=1}^{a-1} \sum_{k \in Z} D_k^{i,j} \Psi_0(a^j x - k)$$

where $\{C_k^j\}_{k \in Z}, \{D_k^{i,j}\}_{k \in Z}, i=1,2,\Lambda, a-1, j = N-M, \Lambda, N-1$ are constant vectors with dimension t . Using Eq.(4), we can decompose $g_j(x)$ as

$$\begin{aligned}
g_j(x) &= \sum_{i=1}^{a-1} \sum_{k \in \mathbb{Z}} D_k^{i,j} \Psi_i(a^j x - k) \\
&= \sum_{i=0}^{a-1} \sum_{k \in \mathbb{Z}} D_k^{i,j} \sum_{m=0}^{a-1} \sum_{l \in \mathbb{Z}} P_{k-al}^{(m)*} \Psi_{ai+m}(a^{j-1}x - l) \\
&= \sum_{i=0}^{a^2-1} \sum_{l \in \mathbb{Z}} \left(\sum_{k \in \mathbb{Z}} D_k^{i,j} P_{k-al}^{(i-\lfloor \frac{l}{a} \rfloor)^*} \right) \Psi_i(a^{j-1}x - l) \\
&= \sum_{i=0}^{a^2-1} \sum_{l \in \mathbb{Z}} D_l^{i,j,1} \Psi_i(a^{j-1}x - l) \\
&= \Lambda \\
&= \sum_{i=0}^{a^{m+1}-1} \sum_{l \in \mathbb{Z}} D_l^{i,j,m} \Psi_i(a^{j-m}x - l)
\end{aligned}$$

where

$$D_l^{i,j,m} = \sum_{k \in \mathbb{Z}} D_k^{i,j,m-1} P_{k-al}^{(i-\lfloor \frac{l}{a} \rfloor)^*}, \quad D_l^{i,j,0} = D_l^{i,j} \quad (5)$$

On the other hand, we can reconstruct $g_j(x)$ as

$$\begin{aligned}
g_j(x) &= \sum_{i=0}^{a^{m+1}-1} \sum_{l \in \mathbb{Z}} D_l^{i,j,m} \Psi_i(a^{j-m}x - l) \\
&= \sum_{i=0}^{a^{m+1}-1} \sum_{l \in \mathbb{Z}} D_l^{i,j,m} \sum_{k \in \mathbb{Z}} P_k^{(i-\lfloor \frac{l}{a} \rfloor)} \Psi_{\lfloor \frac{l}{a} \rfloor}(a^{j-m+1}x - k) \\
&= \sum_{i=0}^{a^{m+1}-1} \sum_{k \in \mathbb{Z}} D_k^{i,j,m-1} \Psi_i(a^{j-m+1}x - k) \\
&= L \\
&= \sum_{i=1}^{a-1} \sum_{k \in \mathbb{Z}} D_k^{i,j} \Psi_i(a^j x - k)
\end{aligned}$$

where

$$D_k^{i,j,m-1} = \sum_{n=0}^{a-1} \sum_{l \in \mathbb{Z}} D_l^{i,j,m} P_k^{(n)} \quad (6)$$

Now, we get Eq. (5) and Eq. (6) as the formula of decomposition and reconstruction of signal function with orthogonal multiwavelet packets with scale=a. They are practical and can be implemented easily.

4. REFERENCES

- [1] Goodman T N T, Lee S L, Tang W S., "Wavelet in wandering subspaces", Trans Amer Math Soc, Vol. 338, No. 3, 1993, pp.639-654.
- [2] Geronimo J, Hardin D P, Massopus, "P. Fractal functions and wavelet expansions based on several scaling functions", J Approx Theory, Vol.78, No.2, 1998, pp.373-401.
- [3] Marasovich J. Biorthogonal multiwavelets, Dissertation Vanderbilt University, Nashville: TN, 1996.
- [4] Coifman R R, Wickerhauser M V, "Entropy based algorithms for best basis selection", IEEE Trans Inform Theom, Vol.32, No.4, 1992, pp.712-718.
- [5] Coifman R R, Meyer Y, Quake S R, *et al*, Signal processing and compression with wavelet packets. in "progress in wavelet Analysis and Applications (Toulouse, 1992)" (Y.Meyer and S.Roques.Eds.), pp. 77-93, Frontieres, Gif, 1993.
- [6] Chui C K, Li C. "Nonorthogonal wavelet packets", SIAM J Math Anal, Vol.24, No.5, 1993, pp.712-738.
- [7] Nielsen M. "Size Properties of Wavelet Packets", ph.D.thesis, Washington University, St.Louis, 1999.
- [8] Nielsen M. Size properties of wavelet packets generated using finite filters. Rev. Mat. Iberoamericana, to appear.
- [9] Nielsen M. Highly nonstationary wavelet packets. Appl Comput Harmon Anal, Vol.12, No.3, 2002, pp.209-229.
- [10] Yang Shouzhi, Cheng Zhengxing, "Orthogonal multiwavelet packets", CSIAM J Appl Math, Vol.13, No.1, 2000, pp.61-65.
- [11] Leng Jingsong, Cheng Zhengxing, Huang Tingzhu, "Biorthogonal multiwavelet packets", CSIAM J Engin Math, Vol.18, No.s1, 2001, pp. 125-130.
- [12] Leng Jingsong, Cheng Zhengxing, "The Properties of Biorthogonal Multiwavelet Packets", The proceedings of the Third International Conference on Wavelet Analysis and its Applications, New Jersey: World Scientific, 2003, pp.484-489



Leng Jinsong is a lecture of School of Applied mathematics of University of Electronic Science and Technology of China. He is now pursuing Ph.D. at Faculty of Sci., Xi'an Jiaotong University. He has published over 10 Journal papers. His research interests are wavelet analysis and signal compression.



Huang Tingzhu is a Professor, Dr. and the Dean of School of Applied Math of University of Electronic Science and Technology of China. He graduated from Dept of Math, Xi'an Jiaotong University in 2001 with Ph.D.. He was a Visiting Professor of Appl. Math Institute of Chinese Science Academia (1999), a Visiting Professor of Hong Kong Baptist University (2002). He has published several books and over 80 journal papers. His research interests are scientific computing, numerical algebra, and matrix analysis with applications etc.

Distributed Cluster-based Solution Techniques for Dense Linear Equations*

Gu Zhimin¹ and Marta Kwiatkowska²

¹School of Information Science and Technology, Beijing Institute of Technology
Beijing 100081, P. R. China
Email: zmgu@x263.net

²School of Computer Science, University Of Birmingham,
Birmingham B15 2TT, United Kingdom
Email: M.Z.Kwiatkowska@cs.bham.ac.uk

ABSTRACT

In many applications, very large matrixes need be solved, however single or multiprocessor system have some limitations on computing resources, this problem was not solved better. This paper discuss a distributed cluster-based solution for dense linear equations, our works included the definitions of notations, Partition of matrix, communication mechanism, improving of the Guassian Elimination and a master-slaver algorithm etc., the computing cost is $O(n^3/N)$, the memory cost is $O(n^2/N)$, the I/O cost is $O(n^2/N)$, and the communication cost is $O(N*n)$, here, n is the grade of matrix, N is the number of computing nodes or processes. We show that our distributed cluster-based solution techniques for dense linear equations could solve the double type of Matrix under 10^6*10^6 effectively.

Keywords: Gaussian Elimination, Cluster-based distributed computing

1. INTRODUCTION

Let A be an $n \times n$ real matrix, let B be a vector in R^n , and consider the system of linear equations $Ax = B$, where x is an unknown vector to be determined [1]. There are many solving methods such as serial software LINPACK etc. [7,8], usually classified as direct and iterative, Direct methods find the exact solution with a finite number of operations, typically the order is $O(n^3)$, Iterative methods do not obtain an exact solution in finite time, but they converge to a solution asymptotically. In many applications such as computational fluid dynamics and weather prediction, as well as image processing and state of Markov Chain etc., n is often very large, and any serial algorithm can't solve the problems. Parallel and distributed computation is currently an area of intense research activity too, motivated by a variety of factors, the computer cluster is generating interest in new types of problems that were not addressed in the past. This paper discuss a cluster-based linear equations solver, our objectives is to construct a direct parallel algorithm for very large n.

Gaussian elimination is the classical procedure in solving linear equations whereby each variable, say the ith variable x^i , is expressed as a function of the variables x^{i+1}, \dots, x^n and is eliminated from the system. After n-1 such steps, the single variable x^n is left and is easily solved. A linear equation group

$AX=B$ could be described into matrix [A B], Gaussian elimination change [A B] into main-diagonal elements of all 1 with some basic shifting and other elements under main-diagonal of all 0. The eliminating procedure need n-1 steps, in the kth step ($k=1,2,\dots,n-1$), the main element, that is the max absolute value selected from $a_{kk}^{(k-1)}, a_{k+1,k}^{(k-1)}, \dots, a_{nk}^{(k-1)}$, then finish basic shifting to [A B] with the main element row. In general, the serial algorithm includes four steps: (1) read data from file to matrix [A B], the cost is $O(n^2)$; (2) $k=1$; (3) while $k < n$ do { select main element from k column; according to some conditions exchange the row of main element with the kth row each other; to the elements of non-main rows do basic shifting by Gaussian elimination; $k=k+1$; } ; (4) print result. The cost of algorithm is $O(n^3)$, and the Memory cost is $O(n^2)$, we need develop an algorithm of parallel linear equations to solve the problem of very large matrix which can not run under resource limitations in single or multiprocessor computer. We present four contributions in this study. (1) We present a partition method of Matrix. (2) We present an adaptive communication mechanism, and show flexible and effective. (3) We present a master-slave Gaussian elimination algorithm. (4) Conducting large matrix simulation, we show our method can solve double matrix under 10^6*10^6 . We discuss the cluster-based parallel solution about serial Guassian algorithm in detail.

2. BASIC PARALLEL ALGORITHM

We use Master-Slave computing model in a computer cluster system, a master program controls all of computing procedures, and other nodes run a slave program only, each slave program have communication with the Master program only. The Linux cluster of Computer School in University of Birmingham, consists of a master node and 22 client nodes, it is connected to the school's network by master. The Master node, named cluster1, dual AMD Athlon 1.6GHz CPU with 256KB cache and 2GB memory is directly accessible from the School's network via the usual methods (SSH, xon, rsh, rlogin, telnet etc) and behaves in a very similar way to the standard linux configuration. The cluster has a local disk mounted as /data/common which contains writable directories for staff & research students. This file system is accessible by the client nodes and can be used to store programs and data for use on the cluster. The client nodes, named cluster1-01 to cluster1-22, dual Athlon 1.6GHz CPU

* This paper is supported in part by CSC of China under grand 21307D05 and by fundament science foundation of BIT and by the university of Birmingham.

with 256KB cache and 1GB memory, reside on a 100Mbps private network and are only accessible from the master node. In addition to having access to the master nodes /data/common file system each node has a private disk which is mounted as /data/private/. This file system may be used to store temporary results from programs running on a node, however, since it is not backed up permanent results should be copied back to the master node or to users' home directories. This Cluster architecture is fit in with a parallel computing in Coarse or medium granule.

2.1 Definitions Of Primary Notations and Variables

Let n be the grade of matrix, let N be the number of computing nodes, and let k is a loop control variable; We assume *client*[N] to be an array of socket value, *localflag*[$n/N+1$] to be an array of row flag for a *part*[$n/N+1$][$n+2$]. For sharing some data, we need a *GlobalInfo* structure, that is, “*typedef struct GlobalInfo{ Major globalmajor; Int k; }*”, here, the definition of *Major* structure is “*typedef struct Major{ int row; double value; double line[n+2]; }*”.

2.2 Partition Of Matrix

According to N which is the number of computing nodes, the matrix $[A \ B]$ is divided into N blocks that size is n/N , and the N partitions are the sets of row numbers $(1, \dots, n/N), (n/N+1, \dots, 2*n/N), \dots, ((N-1)*(n/N)+1, \dots, n)$, each one of the blocks be only assigned to one of the N computing nodes or processes. There is a limitation which is the max number of partitions to be less or equal n . In general, the matrix $[A \ B]$ is in a file, after partitioned, we locate for the beginning of the first row of each block with the function *ChildNo*(n/N)*((n+1)*11)*, here, *ChildNo*=0, 1, ..., $N-1$, a value of *ChildNo* get by the function *GetChildNo()*. Each partition get the data of matrix *part*[][] with *Void input(double part[n/N+1][n+2], int childNo)*. Meanwhile, to some row or column, we could add some 0 into $[A \ B]$ under some cases, this could maintain a partition into real requirement. When the computing nodes are monopolized by a task, we implement basic load balance of a matrix $[A \ B]$ in our Cluster, because all computing nodes in the Cluster system have same architecture and operating system.

2.3 Communication Mechanism

Communication technology plays an important role in a parallel computation, and might be a potential bottleneck [2]. We are focus on parallel communication used in master-slave model, and propose a simple protocol to provide the communication service for some intensive application of data. Why do we need new communication model rather than MPI or PVM? The reason is just the communication interface can serve our parallel applications more flexible and effective, and is extended to Network-based or Gird-based computing easily. We use the TCP/IP protocol as a basic transport mechanism between computing nodes, and have developed a new Communication interface that is easily used in the parallel computing of linear equations. They includes *Safe_send(int socket, void* msg, int totallen)*, *Safe_recv(int socket, void* buf, int totallen)*, *Multicast(void* msg, int len, int client[N])* and *Multicast_childNo(int client[N])*.

2.4 Improving Guassian Elimination Method

The traditional guassian elimination method changes matrix $[A \ B]$ into all 1 main-diagonal elements and all 0 other elements under main-diagonal, we use this serial method and improved it, the revised guassian elimination method could

change matrix $[A \ B]$ into all 1 main-diagonal elements and all zero other elements in matrix A , we could just get the computing results from matrix B , and the cost of function *guass()* is $O(n^2/N)$.

2.5 Distributed Cluster-based Computing

The computing model is a typical Master-slave architecture, we describe the algorithm of Master and slave respectively. According to the communication relations of sending and receiving in distributed cluster-based computing for linear equations, we get the formula about the communication cost, that is, $Comm_{Cost} \approx (N-1)+(N-1)+n*((N-1)+(N-1)+2)+(N-1) = 2n*N+3N-3 = O(N*n)$.

The Algorithm of Master:

1. Create socket, bind and listen;
2. $i=1$;
3. if $N=1$ then break;
4. receive the value of connecting socket to *client*[$i++$] with *accept*; if $i=N$ then break else goto 4;
5. *multicast_childNo(client)*;
6. $k=1$
7. *input(part,0)*;
8. for ($i=1$; $i \leq n/N$; $i++$) *localFlag*[i]=0;
9. *index=1*; *change=0*;
10. *change*[$++change$]= k ; *localmajors*[0].*value*=0;
11. select local max value from *part*[j][*index*], and *localmajors*[0].*value*=*part*[j][*index*]; *localmajors*[0].*row*= j ;
12. receive *localmajor*[i] from all slave with *safe_recv*; choose the max value to *gingo.globalmajor.value* and the value of i to *majorchildNo*;
13. if the max value < infinitesimal then exit(1);
14. *gingo.globalmajor.row*=*majorChildNo*(n/N)+localmajors[majorChildNo].row*;
15. for ($i=1$; $i \leq n+1$; $i++$)
gingo.globalmajor.line[i]=*localmajors[majorChildNo].line*[i]
/gingo.globalmajor.value;
16. *gingo.k=k*;
17. *multicast GlobalInfo* to *client*[];
18. if $!(k \text{ in } local)$ and $!(gingo.globalmajor.row)$ and $!(inSameChild(gingo.globalmajor.row,k))$ then { receive the values of the k th row from *client*[*getChildNo*(k)] with *safe_recv*; send the values to *client*[*getChildNo*(*gingo.globalmajor.row*)] with *safe_send* }
19. if $k \text{ in } Local$ and $!(gingo.globalmajor.row)$ then {send the *part*[k] row to *client*[*getChildNo*(*gingo.globalmajor.row*)] with *safe_send*;
*memcpy(part[k],gingo.globalmajor.line,(n+2)*sizeof(double))*;
localFlag[k]=1; }
20. if $!(k \text{ in } Local)$ and *gingo.globalmajor.row* {receive the values of k th row with *safe_recv*; put them to *part*[*gingo.globalmajor.row*] }
21. if $k \text{ in } Local$ and *gingo.globalmajor.row* then {use *memcpy* to exchange the values of both *part*[k] and *part*[*gingo.globalmajor.row*]; *localflag*[k]=1; }
22. use *gauss(part,gingo.globalmajor.line,index,k,kinlocal)* to computing;
23. if $k=n$ then break
24. $kk=k+n/N$;
25. if $kk>n$ then $k=(kk+1)\%n$ else $k=kk$;
26. *index=index+1*; if *index* <= n then goto 10;
27. receive the result of *part*[][][$n+1$] from all slaves with *safe_recv*;
28. print the results.

The Algorithm of Slave:

1. Create a socket, and have a connection with Master
2. For ($i=1$; $i \leq n/N$; $i++$) *localFlag*[i]=0;
3. Receive *childNo* With *safe_recv*, then accept the Local data *part*[][] from *MatrixFile* with *input*;
4. *index=1*;

5. *Select local max value of part[j][index], and localmajor.value=part[j][index], localmajor.row=j;*
6. *Send the value of localmajor to Master with safe_send;*
7. *Receive the value of globalInfo from Master with safe_recv ;*
8. *If k in local and !(ginfo.globalmajor.row in local) then { send part[getLocalIndex(k)] to Master with safe_send; memcpy(part[getLocalIndex(k)], ginfo.globalmajor.line, (n+2)*sizeof(double));localFlag[getLocalIndex(k)]=1;}*
9. *If ginfo.globalmajor.row in local and !(k in local) then { put the values of the kth row to buf with safe_recv; memcpy(part[getLocalIndex(ginfo.globalmajor.row)], buf, (n+2)*sizeof(double));}*
10. *If k in Local and ginfo.globalmajor.row in local then{memcpy(part[getLocalIndex(ginfo.globalmajor.row)],part[getLocalIndex(k)],(n+2)*sizeof(double));memcpy(part[getLocalIndex(k)],ginfo.globalmajor.line,(n+2)*sizeof(double));localFlag[getLocalIndex(k)]=1}*
11. *Guass(part, ginfo.globalmajor.line, index, k, kInLocal);*
12. *Index=index+1, if index <=n then goto 5;*
13. *Send the result part[[n+1]] to Master with safe_send*

The computing cost in Master is $O(n^3/N)$, and the memory cost in Master is $O(n^2/N)$; the computing cost in Slave is $O(n^3/N)$ too, and the memory cost in Slave is $O(n^2/N)$ too; the I/O cost in Master is $O(n^2/N)$, same in slave; the communication cost is $O(N*n)$. For effective parallel computing, the solution must have some limitation, such as $TIME_{communication} \leq TIME_{computing} + TIME_{memory}$ for each computing process. If N is n^2 , the computing cost in Master and Slave are all $O(n)$, and the memory cost is a constant, we point out that is not fit in with the solution of coarse or medium granule. If N is n , the computing cost in Master and Slave are all $O(n^2)$, and the memory cost is $O(n)$, but the cost of communication is $O(n^2)$. In general, n is a constant, when choosing N , we need have good trade off for real time performance. We notice the following relations too: (a) to $O(N*n)$, the communication cost will get higher lineally when N get more gradually; (b) to $O(n^3/N)$, the computing cost will get lower fast when N get more gradually; (c) to $O(n^2/N)$, the Memory cost will get lower fast when N get more gradually.

3. DATA SIMULATION of MATRIX[AB]

For test our programs, we construct a simulator of Matrix [A B], its grade could be any size. The simulation procedure is as following: First, according to the grade n , the array answer[] is created by using *drand48()* which is an random function; Second, according to this array answer[], create a matrix [A B]; at last, the matrix [A B] be written in the DATAFILE.

4. THE RATIONALE of IMPROVING PARALLEL ALGORITHM

When n and N are very high, we will meet some challenges on some limitations of max array space in memory and max file space in hard disk. For example, when n is 10^9 , if each data is a double type, need about $64*10^9$ Gb size of array memory space. This space limitation in memory of our computing node is about 1GB, and we could only use 64MB size of array space in each computing process according to our test, and the file space is very big too. The limitations will affect the structures and performance above algorithms. We have to

adopt some new computing methods to trade off between memory and disk.

We let S_l be the block size of memory limited for each process, $S_{each-row}$ be the size of each row, $S_{n/N}$ be the size of each col in a block, $S_{n2/N}$ be the size of any block. For any block of $((i-1)*n/N+1, \dots, i*n/N)$, if $(S_{n2/N} \leq S_l)$ and $(S_{each-row} < S_l)$ are ture, above basic parallel algorithm meet our requirement, however, when n and N are very high, we have to meet the number of TCP connections and modified our programs, and Master is only used to communicate and show computing results, and do not do any computing of *part[][]*, here, the computing cost in Master is $O(n)$, cost of I/O is $O(n)$ too; Costs of slave and cost of communication are same above; and $S_l=64$ MB, N is only the number of computing processes which are in some slave nodes. Our revised algorithms could solve the double type matrix under 10^6*10^6 .

However, if $S_{n2/N} > S_l$ is ture, above basic parallel algorithms do not meet this requirement, we must improve the algorithms again.

- (1) When $(S_{each-row} > S_l)$ and $(S_{n2/N} > S_l)$ are ture, we cut $S_{each-row}$ with S_l , and finish $S_{each-row}/S_l$ times for each row, the total times are $int(n/N) * S_{each-row}/S_l$ here, x denotes $int(x)(x-int(x)=0)$ or $int(x)+1(x-int(x)>0)$;
- (2) When $(S_{each-row} = S_l)$ and $(S_{n2/N} > S_l)$ are ture, we need $int(n/N)$ times for a black;
- (3) When $(S_{each-row} < S_l)$ and $(S_{n2/N} > S_l)$ are ture, we cut S_l with $S_{each-row}$, and need to finish $S_l/S_{each-row}$ times for each S_l , the total times are $S_l/S_{each-row} * int(n^2/N)/S_l$.

Meanwhile we can improve the method of choosing max element in any block too, key algorithms are as following:

- (1) When $(S_{n/N} < S_l)$, we choose only a time.
- (2) When $(S_{n/N} = S_l)$, we choose only a time too.
- (3) When $(S_{n/N} > S_l)$, we choose $(S_{n/N})/S_l$ times.

We could decompose matrix file [A B] into N subfiles such as $[A1 B1], \dots, [A_N B_N]$, here, the $[A_i B_i]$ is just the block $_i$. However, we know that I/O cost is very higher than memory. According to our test to basic I/O in Preston workstation, the costs of open-a, open-w, open-r, w/a and r are 0.0008s, 0.0013s, 0.0006s, 0.0004s and 0.00002s respectively. If 10^9 double type of data are written to some file, need $0.0004s*10^9=400000s=400000/3600h=111.1h=111.1/24$ days=4.625 days at least, this is a challenge for continuous writing in file system of multiple users. Anyway, this is a difficult problem for very large n .

5. SOME EXAMPLES

We know $T2$ is a good result in Table1, the number of slaves with $T1$ was less, but the number of slaves with $T3$ and $T4$ were more. Here, 179.8s* in $T5$ was gotten by four processes in one computing node; 259.7s* in $T6$ is gotten by nine processes in one computing node. $T7$ denotes 16 or 8 or 4 or 2 computing nodes and 1 master(no part[][]), the 980.0s* was got by 17 processes in one computing node; $T8$ denotes 4 or 7 computing nodes and 1 master(no part[][]), the 1675.2s* was got by 26 processes in the Cluster1. We have further analysis: $T7$ gives the max N is 16 with 1Gb memory in a computing node, could only compute a $4000*4000$ double type matrix,

T9 gives the max N is 361 when each node in our cluster of 22 nodes run 16 processes, could only compute a 18500×18500 double type matrix. T10 gives the max N is 10^6 when computing a $10^6 \times 10^6$ double type matrix, need 64000Gb memory, further we could know that need $10^6/16=62,500$ computing nodes. For the problem, hundreds of our Cluster could not meet the requirement, it is an internet-based computing.

Figure 1 shows our test results: when n get higher, memory size, computing size, number of slave processes and traffic size get higher together, see fig.1 (1), (2), (3) and (5); and our parallel computing size get higher very slowly, but the 100MB/s communication rate in our cluster system affect our test performance, see fig.1 (4). Fig.1 (6) shows the distributions of data and results in different computing nodes.

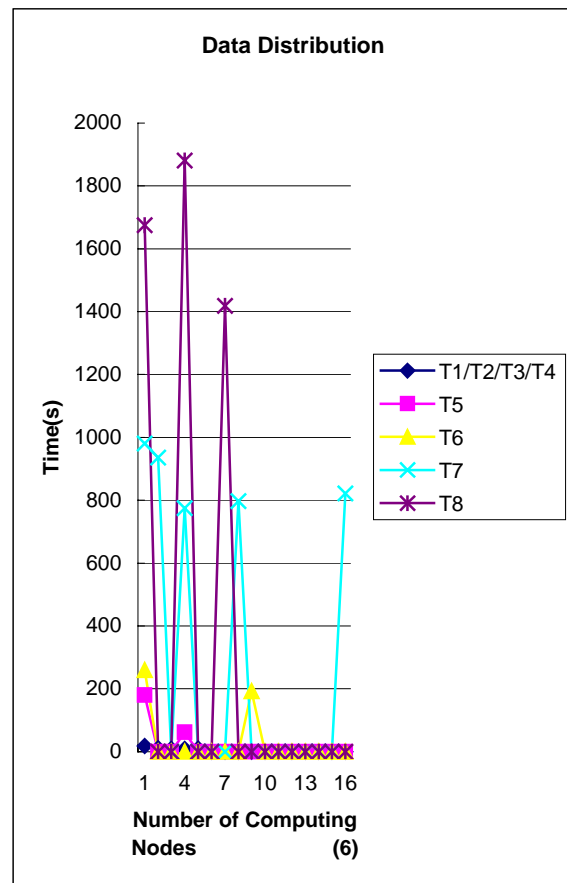
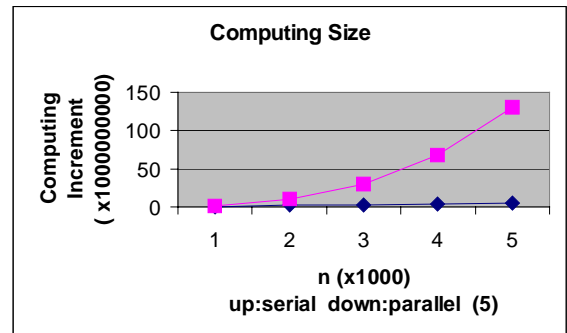
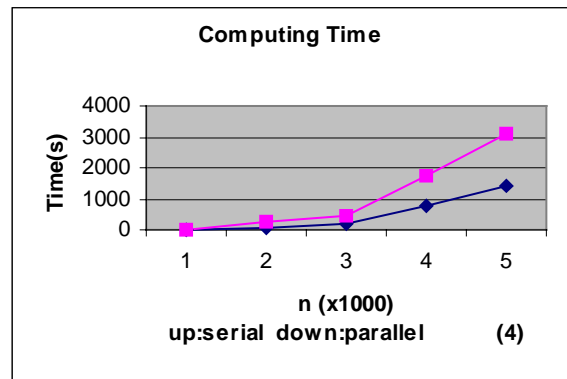
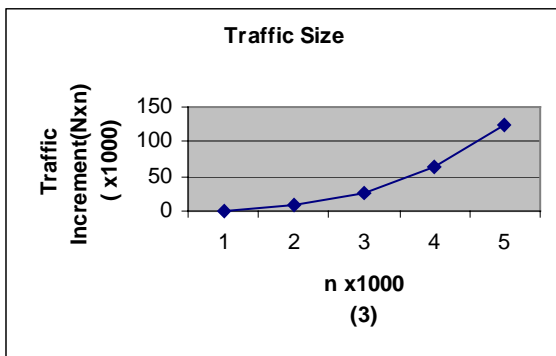
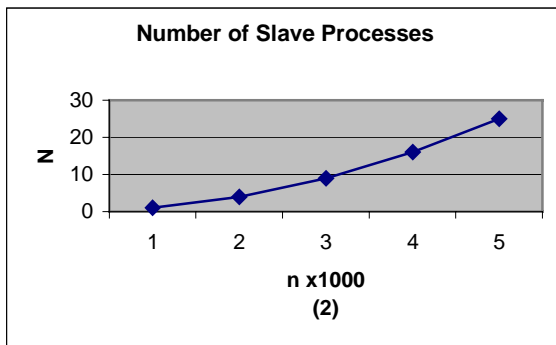
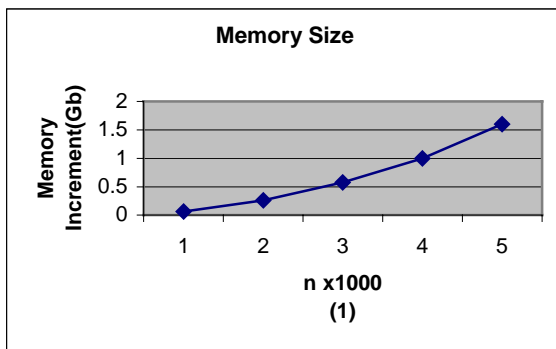


Figure 1 Test results with n get higher, memory size, computing size, number of slave processes and traffic size get higher together.

(1) Memory Size (2) Number of Slave Processes

- (3) Traffic Size (4) Computing Time
(5) Computing Size (6) Data Distribution

6. CONCLUSION

The master-slave computing model is powerful and effective in a Cluster system for dense linear equations problem under $10^6 \times 10^6$ of double type data, and have good future in Network-based or Internet-based computing or grid computing. The important data of different computing nodes could effectively be shared by the GlobalInfo structure. The communication mechanism based on TCP used in both master and slave could work very well. The partition strategy in our distributed system shows rather flexibility too.

Acknowledgments

This work is supported in part by CSC of China under grand 21307D05 and by fundament Science Foundation of BIT and the University of Birmingham. The authors would like to thank PhD candidates Rashid Mehmood and Jiang Ming 'help for LINUX Cluster system and his suggestions. Junchang Ma and Huifang Cheng participated in the work in the early stage.

7. REFERENCES

- [1] Dimitri P. Bertsekas, John N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods. Athena Scientific, Belmont, Massachusetts, 1997.
- [2] Douglas E. Comer, David L. Stevens, Internetworking with TCP/IP vol III: Client-Server Programming and Applications, BSD Socket version, 2nd ed. Prentice Hall, Inc. 1996.
- [3] Megson G M, Evans D J, Improved matrix product computations using double-pipeline systolic arrays, the Computer Journal, Dec.1988, 31(6):567-570
- [4] Hwang K, Cheng Y H, Partitioned matrix algorithm for VLSI arithmetic system, IEEE Trans. On Computer, Dec.1982,C-31(12):1215-1224
- [5] Daniel D. Deavours and William H. Sanders, An Efficient Disk-based Tool for Solving Large Markov Models. Performance Evaluation, 33(1):67-84, 1998.
- [6] Daniel D. Deavours and William H. Sanders, An Efficient Disk-based Tool for Solving Large Markov Models. In Lecture Notes in Computer Science: Proceedings of the 9th International Conference on Modelling Techniques and Tools(TOOLS'97), pages 58-71, St. Malo, France, June 1997. Springer-Verlag.
- [7] Chi Xuebin, Parallel Solving Linear Systems On A Hierarchy Memory Multiprocessor, May 1995,210-217.
- [8] J.J.Dongarra, C.B.Moler,J.R.Bunch,G.W.Stewart, LINPACK User's Guide, SIAM Philadelphia,1979.



Gu Zhimin received the BS in Computer Science from Shanxi University, China, in 1985, and the MS in computer science from Harbin Institute of Technology, China, in 1991, and the PhD in computer science from Xian Jiaotong University, China, in 1997. He is a professor of Computer Science at Beijing Institute of Technology and visiting scholar of Computer science at the University of Birmingham. His research interests are in the areas of distributed and internet systems, computer architecture, and science computing.

Marta Kwiatkowska am Professor of Computer Science in the School of Computer Science at the University of Birmingham. Her main research interests are modelling and verification of probabilistic and real-time systems (theory and implementation); model checking; semantics of concurrent and distributed systems; and modal and temporal logics.

TABLE 1 Performance Tests

Task Name	n (64 bits)	N (Number of Processes or Nodes)	Memory Cost	Parallel Time(s)	Serial Time(s)	Speedup
T1/T2/T3/T4	1000	2/3/4/5	64M	11.0/8.5/9.1/9.5	18.2	1.65/2.14/2.0/1.92
T5	2000	4	256M	61.9	179.8*	2.89
T6	3000	9	576M	192.2	259.7*	1.35
T7	4000	1/16Slaves 2/8S 4/4S 8/2S	1G	821.48 797.36 775.15 935.20	980.0*	1.19 1.23 1.26 1.05
T8	5000	8/3S+1/1S 4/6S+1/1S	1.6G	1880.80 1418.21	1675.2*	0.89 1.18
T9	19000	361	23.1G	-	-	-
T10	10^6	10^6	64000G	-	-	-

Apply Neural Computation to Ground Waves Caused by High-Speed Trains

Zou Chengming, Yang Hongyun, Tong Qiwei, Zhong Luo
The Computer Science & Technology Department, Wuhan University of Technology
Wuhan, Hubei, 430070, China
Email: {zouc, yhy}@mail.whut.edu.cn, Tel: 027-87211983

ABSTRACT

In this paper, finite elemental problems are transmitted into a quadratic programming by using the minimum potential energy theorem. A method, which can be applied to the quadratic programming, is that an optimization problem can be mapped into a dynamic circuit by using proper neural network, and it can converge to the global minimum by using genetic algorithm within circuit times.

Keywords: Neural Computation; Genetic Algorithm; Finite Element; Ground Waves

1. INTRODUCTION

Neural Network is a complex nonlinear mechanics system with high parallel computational ability. By using a proper neural network, an optimization computation problem can be mapping into a dynamic circuit, and get the result at the scale of circuit times. It may be possible to fulfill the modern structure analysis timely and modulate analysis of complicated mechanics actives if the finite element computation of mechanics can be described into the equality constrained optimization problem.

2. QUADRATIC OPTIMIZATION COMPUTATIONS

The equality constrained quadratic optimization problem can be described:

$$\begin{cases} \min \varphi(x) = \frac{1}{2} x^T G x + C^T x \\ \text{s.t. } A_i^T x = J_i \quad i = 1, 2, K, m \end{cases} \quad \text{Eq(1)}$$

In the equation set, x R_n is a optimization vector, C and A_j R_n are constant vectors.

Eq(1) describing the equality constrained quadratic optimization problem can be completed by TH neural network.^[1]

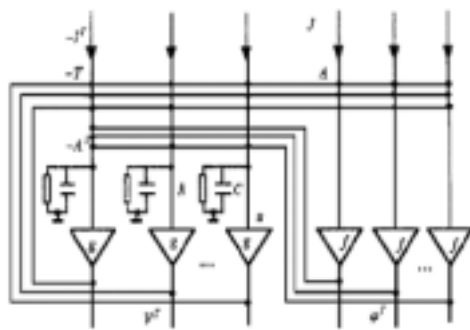


Fig.1 The TH Neural Network Constitution

It includes two parts. The left is signal network. The right is constraint network. Every symbol's meaning is given below:

- f : A enlarger function of equality constrained plane. $f(y)=\alpha y$, and α represents resistor, y represents current.
- g : signal enlarger function, $V_i = g(u_i)=\beta u_i$ and β represents scalar.
- V R_n : signal network output voltage vector, corresponding the optimization vector x in the quadratic optimization Eq(1).
- I R_n : signal network input vector, corresponding the once coefficient CT in the quadratic optimization Eq(1).
- J R_m : constraint network input vector, corresponding the constant vector J in quadratic optimization Eq(1).
- G : self-jumper matrix, corresponding the matrix G in Eq(1).
- A : different jumper matrix, corresponding the coefficient matrix A of the constraint equation in Eq(1).
- ϕ R_m : output voltage vector of constraint network.

According to KCL(Circuit Theorem),the dynamic equation of the quadratic optimization network can be obtained:

Signal network :

$$C_i \frac{du_i}{dt} = -I_i - \frac{u_i}{R_i} - \sum_{j=1}^n g_{ij} V_j - \sum_{j=1}^m a_{ji} \phi_j \quad i = 1, 2, K, n \quad \text{Eq(2)}$$

Constraint-network

$$\phi_i = F(-J_i + \sum_{j=1}^m a_{ij} V_j) \quad i = 1, 2, K, m \quad \text{Eq(3)}$$

The energy function can be defined:

$$E(V) = I^T V + \frac{1}{2} V^T G V + \sum_{j=1}^m F(A_j^T V - J_j) + \sum_{i=1}^n \frac{1}{R_i} \int_0^V i g^{-1}(\tau) d\tau \quad \text{Eq(4)}$$

F : uncertain integral, $F(y) = \frac{\alpha}{2} y^2$

As Eq(4) shows, the first and the second items represent the cost function of quadratic optimization, the third represents the measurement contravention constraint condition, and the fourth item serves as adjustment. The purpose of optimization is to find the number, which makes value of the third item be null and the first, the second item have the minimum (the fourth can be omitted). Then we can obtain the equation

$$\frac{dE}{dt} = \sum_{i=1}^n dV(I_i + \frac{u_i}{R_i} + \sum_{j=1}^n g_{ij}V_j + \sum_{j=1}^m a_{ji}\varphi_j) = \sum_{i=1}^n \frac{C_i}{\beta} (\frac{dV_i}{dt})^2 = -\frac{C}{\beta} \|\frac{dV}{dt}\|^2$$

$$\ominus \quad C_i, \beta > 0,$$

$$\therefore \text{Only when, } \frac{dE}{dt} \leq 0 \text{ and } \frac{dV}{dt} = 0, \quad \frac{dE}{dt} = 0 \quad \text{Eq (5)}$$

So $E(V)$ is the Lyapunov function of the neural network system described by Eq (2) and Eq (3). The Eq (5) shows: the system energy function will be diminished on and on while the time builds up, and until the whole system reaches the stable state, it is corresponding the local minimal solution. Thus this is the local minimum of the equality constrained quadratic optimization problem. It can be proved that when matrix G is positive definite, the local minimum is the global minimal solution. But in actual works, the matrix G transforming a practical works problem into quadratic optimization form is semi-positive definite. At the time getting the global minimum solution, we need search the global minimum using the global optimization algorithm. Then the following will be discussed in the problem how to make the system convergence to the global minimum by taking advantage of genetic algorithm.

3. GENETIC NEURAL OPTIMIZATION

From Eq(2), the stable state(balanced point) of the neural network is fit for^[1]:

$$I_i + \frac{u_i}{R_i} + \sum_{j=1}^n g_{ij}V_j + \sum_{j=1}^m a_{ji}\varphi_j = 0 \quad \text{Eq(6)}$$

In Eq(6), V_i is an unknown variable, it can be remarked:

$$F(V)=0 \quad \text{Eq(7)}$$

In the formula,

$$V = [V_1 \ V_2 \ \dots \ V_n]^T. \quad \text{Eq(7)}$$

is a high nonlinear algebraic equation, and it is greatly difficult to find the accurate solution, even the solution does not exist. In order to solve the problem, the global optimization algorithm--the genetic algorithmic can be applied to get the receivable solution[2.3.4]. The genetic algorithmic is:

- 1) coding : The method of real coding is applied. Since it is a nonlinear problem, real coding not only express naturally, but also enhance the computational precision.
- 2) fitness function : In order to assure that $f(x)$ is not negative, select $f(x)=|F(x)|$ as fitness function.
- 3) selection strategy : The purpose of selection is to genetic the optimization individual to the next generation or genetic the new individual created by doubled crossing to the next generation. The selective probability of generation K 's number I is given below:

$$P_{k(i)} = \frac{f_{k(i)} - f_k}{\sum_{j=1}^N (f_{k(j)} - f_k)}, \quad f_k = \min\{f_{k(i)}, i=1, K, N\}$$

In the formula, N is the species group scale, $f_{k(i)}$ is the adaptability of the individual I of generation k 's. After completing selection, cross, crochot, if the best individual q of generation K is not in the next generation, it will be as number $n+1$ of generation $k+1$.

- 4) crossover strategy : Crossover is the operation that replaces and reassembles the component structure of the two father generations' individuals into new individuals. The operation can be defined:

$q_1 = (V_1^1, V_2^1, K, V_m^1)$ and $q_2 = (V_1^2, V_2^2, K, V_m^2)$ are two solution vector of father generation, while $q_3 = (V_1^3, V_2^3, K, V_m^3)$ and $q_4 = (V_1^4, V_2^4, K, V_m^4)$ are two discordances through crossover. So the general arithmetic crossover is given below:

Firstly, random numbers a_1, a_2, \dots, a_m are created in the range from 0 to 1,

$$V_i^3 = a_i V_i^2 + (1 - a_i) V_i^1 = V_i^1 + a_i (V_i^2 - V_i^1)$$

$$V_i^4 = a_i V_i^1 + (1 - a_i) V_i^2 = V_i^2 + a_i (V_i^1 - V_i^2)$$

$$i=1, 2, \dots, m,$$

In order to simple the computation, $a_1=a_2=\dots=a_m$ is selected.

5) mutation strategy : mutation operation is to change some gene values of individual's chain in group. The purpose is to make the genetic algorithm to have the ability of local random search. When the genetic algorithmic accesses to the best solution domain, mutation can convergence more quickly directed to the best solution. The mutation operation also sustains the variety of the group and averts immature convergence. The operation is given below:

$q = (V_1, V_2, K, V_m)$ is a father generation vector in solution space, $f(q)$ is its adaptive value, f_{\max} is the maximal value for the problem. The mutation temperature is defined as below:

$$T = 1 - \frac{f(q)}{f_{\max}}$$

If V_k is decided to mutated in the defined range is $[a_k, b_k]$,

the solution is $q' = (V_1', V_2', K, V_m')$, here

$$V_k' = \begin{cases} V_k + (b_k - V_k)(1 - r^{\lambda}) & r < 0.5 \\ V_k - (V_k - a_k)(1 - r^{\lambda}) & r > 0.5 \end{cases}$$

R is a random function in the range $[0,1]$; λ is a parameter to decide the mutation degree, whose function is to adjust the local search domain. The range is generally from 2 to 5.

From the genetic algorithmic, if $f(x_1) < f(x_2)$, it shows the individual x_1 is better than the individual x_2 .

Supposing X is the space of all populations P . For given population, $P = \{x_1, x_2, K, x_m\}$ define

$$\text{Eval}(P) = \frac{1}{n} \sum_{i=1}^n \text{eval}(x_i) \quad \text{fitness function of } P.$$

Obviously $\text{Eval}(P) \geq 0$.

A measurement is defined as below:

$$d(P_1, P_2) = \|P_1 - P_2\| = \begin{cases} 0 & P_1 = P_2 \\ \text{Eval}(P_1) + \text{Eval}(P_2) & P_1 \neq P_2 \end{cases}$$

(X, d) is a measurement space[5], which fits:

- 1) For all populations P_1 and P_2 , $d(P_1, P_2) \geq 0$, only when $P_1 = P_2$, $d(P_1, P_2) = 0$.
- 2) $d(P_1, P_2) = d(P_2, P_1)$.
- 3) $d(P_1, P_2) + d(P_2, P_3) = \text{Eval}(P_1) + \text{Eval}(P_2) + \text{Eval}(P_2) + \text{Eval}(P_3) \geq \text{Eval}(P_1) + \text{Eval}(P_3) = d(P_1, P_3)$.

What's more, the measurement space (X, d) is self-contained. Because for all populations, there are finite quantity numbers, and for Cauchy sequence P_1, P_2, \dots of any group, there is a K , when $n > k$, $P_n = P_k$. Thus all Cauchy convergence sequences have limit. So the measurement space (X, d) is a Banach space.

Now define mapping $T : X \rightarrow X$, which is a simple

iteration in genetic algorithmic. The iteration is $T(P(t))=P(t+1)$. Since population $P(t)$ evolve to population $P(t+1)$ along the evolving direction (No count if no improvement), it is to say that

$$\text{Eval}(P(t)) > \text{Eval}(P(t+1)) = \text{Eval}(T(P(t))).$$

So :

$$\begin{aligned} \|T(P_1(t)) - T(P_2(t))\| &= \text{Eval}(T(P_1(t))) + \text{Eval}(T(P_2(t))) \\ &< \text{Eval}(P_1(t)) + \text{Eval}(P_2(t)) = \|P_1(t) - P_2(t)\| \end{aligned}$$

From the above,

$$\|T(P_1(t)) - T(P_2(t))\| \leq \alpha \|P_1(t) - P_2(t)\|, \alpha \in [0, 1]$$

can be gotten. T is a compressible mapping. According to the compressible mapping principle, T has a single fixed point, remarked as P^* .

$$P^* = \lim_{i \rightarrow \infty} T^i(P(0))$$

That is say the compressible genetic algorithm convergent to population P^* , which is the only fixed point in the population space and has nothing with population $P(0)$. According to the definition of $\text{Eval}(P)$, when all individuals have the same global minimization, the fixed point P^* can be obtained.

The whole illustration of genetic algorithmic shows $F(V)$ convergences to the global optimum solution (the minimum).

4. GROUND WAVES ANALYSE BY NEURAL COMPUTATION

During the last decade, high-speed railways have become one of the most advanced and fast developing branches of transport. Unfortunately, the increased speeds of modern trains are normally accompanied with increased transient movements of the rail and ground, which may cause noticeable vibrations. This brings many scholars to study on ground waves caused by moving trains.

When analysing ground waves by finite element method, displacements is often used as the solution of the problem. After getting the displacements of every node, it will be easy to obtain the stress and strain of the cell. Here select the node's displacements as the output of the neural network. After discrete the space domain in finite element method, by using minimum potential energy theorem, the potential energy of the whole structure can be given as below :

$$\Pi = \frac{1}{2} \{\delta\}^T [K] \{\delta\} - \{\delta\}^T \{f\} - [C] \{\delta\} - [M] \{\ddot{\delta}\} \quad \text{Eq(8)}$$

In Eq(8), $\{\delta\}$ is the displacement vector, $\{f\}$ is a load vector, $[K]$ is a stiffness matrix, $[C]$ is a amortization matrix, and $[M]$ is a mass matrix.

Transforming Eq(8) as follows:

$$\Pi = \frac{1}{2} x^T G x - x^T F \quad \text{Eq(9)}$$

Here:

$$x = \begin{bmatrix} \{\delta\}^T & \{\dot{\delta}\}^T & \{\ddot{\delta}\}^T \end{bmatrix}^T$$

$$F = \begin{bmatrix} \{f\}^T & \{0\}^T & \{0\}^T \end{bmatrix}^T$$

$$G = \begin{bmatrix} [K] & [C] & [M] \\ [C] & 0 & 0 \\ [M] & 0 & 0 \end{bmatrix}$$

The solution of the whole problem is actually to get the minimum of formula Eq (9). Thus has the follow equation set:

$$\begin{cases} \min \Pi = \frac{1}{2} x^T G x - x^T F \\ s.t. Ax = \bar{x} \end{cases} \quad \text{Eq(10)}$$

$Ax = \bar{x}$ is the constraint condition of the problem.

To Eq(10), it can be gotten the result by using the genetic neural optimization computation method.

5. CONCLUSION

By using minimum potential energy theorem in finite element method, ground waves problems can be transmit into the equality constrained quadratics optimization problem. It can be get the result by using the method, which is discussed in the paper, the genetic neural optimization.

6. REFERENCES

- [1] Jiao Lichen, neural network computation [M]. the publishing house of Xi'an university of electronic and technology, 1995.
- [2] Chen Guoliang, Wang Xifa, Zhuang Zhenquan etc. Genetic algorithm and its employment [M]. Beijing: the publishing house of people's mail, 2001.
- [3] Zhong Luo, Bai Zhenggang, Xia Hongxia etc., the modulation optimization and its employment of the grey problem about NN. 2001, 9.
- [4] Zhong Luo, The employment research of Neural Network in the expert system exploitation tool at architectural structure design [J]. the paper of master degree in Wu Han university of technology. 1996, 4.
- [5] Li Yun, Modern mathematic method and its employment in nonlinear dynamic system [M]. the publishing house of people's traffic, 1998.
- [6] Sun Daoheng, Hu Qiao, Xu Hao, the real-time neural computational principle and value emulation of elastic mathematics [J]. the studying newspaper of mathematics. 1998,



Zou Chengming is an instructor, who is working at School of Computer Science and Technology, Wuhan University of Technology. He got the doctor degree from Structure Engineering, Wuhan University of Technology in 2003. His research interests are neural network, distributed computation.

A New Distributed Computing Model: Isolated Island

Xinchao Zhao

Key Laboratory of Mathematics Mechanization,
Institute of Systems Science, AMSS, Academia Sinica
Beijing, 100080, China
Email: xczhao@mmrc.iss.ac.cn

ABSTRACT

Inspired from the society and civilization optimization algorithm and the distributed genetic algorithm, a new isolated island computing model is proposed. The definition domain of the variables is divided into many adjacent and exclusive sub-domains, and then the optimum is found at every sub-domain respectively. Finally, the best solutions among all the optimal solutions of all the sub-domains are returned. One obvious feature of the isolated island model is that it is possible to find all the optimal solutions of a given multimodal problem at a single run. The isolated island algorithm based on this model and hill-climbing strategy shows the dominance to the local search through the experimental results. What's more, simulated annealing is also incorporated into the proposed I&I model for some benchmarks and even better results are obtained.

Keywords: isolated island model, distributed computation, society and civilization, local search, simulated annealing, genetic algorithm

2. INTRODUCTION

An isolated island (I&I) distributed computing model is proposed in this paper based on the hill-climbing local search [1, 18]. This idea is inspired from the society and civilization optimization algorithm [2] and the distributed genetic algorithm [3, 4]. As we know, "premature convergence" is a major problem in the use of genetic algorithm, viz, the population has already been homogeneous before the algorithm finds the optimal solution of the problem. The loss of critical alleles due to selection pressure, selection noise, schemata disruption due to crossover operator, and poor parameter setting may make this **exploration/exploitation** balance disproportionate, and produce a lack of *diversity* in the population [5, 6, 7]. Many researches focused on the improvement of operators and parameters settings of GAs, such as [8].

Another approach for dealing with this problem is the distributed genetic algorithm (DGA) [3, 4]. Its basic idea is to keep, in parallel, several subpopulations that are processed by genetic algorithms, with each one being independent of the others. DGA is widely researched [3, 4, 7, 9, 10] and very encouraging results are obtained. But every sub-population of DGA is still to search the solutions of the problem in the total definition domain space. All the sub-populations of DGA maybe converge to the same local traps and DGA may be impossible to find all the optimal solutions at a single run, especially for the multimodal problems. The individuals in [2] are separated into a number of mutually exclusive clusters (society and civilization) based on their *Euclidean distance*. But the initial individuals are still randomly generated in the total definition domain as genetic algorithms do. It is possible

that there are some areas being not explored or exploited from begin to end.

Greatly different from the above two methods, the isolated island model is to artificially divide the definition domain into many appropriately small adjacent and exclusive sub-domains. Subsequently, an initial solution is randomly generated at every sub-domain and a hill-climbing algorithm is used to find the "optimal solution" starting from the initial solution at every sub-domain. Finally, all the "optimal solutions" are collected from all the sub-domains and the best results are returned.

The rest of this paper is organized as follows. In Section 2 the isolated island model is introduced. Some benchmarks are discussed in Section 3. Section 4 we make comparisons between our algorithm and local search by solving some benchmarks. In Section 5, possible parallel strategies for the I&I model are presented. Finally, conclusions are arrived at and future works are discussed in Section 6.

2. ISOLATED ISLAND MODEL

The isolated island model is illustrated based on a two dimensional optimization problem in this Section.

Six-Hump Camel-Back Function [11]

$$(4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$$

where, $-5 \leq x_1 \leq 5$, $\min(f) = -1.0316285$ at $(0.08933, -0.7126)$ and $(-0.08983, 0.7126)$ is selected to demonstrate how our model is implemented. Lower bounds are collected in an array **LB** = $\{-5, -5\}$ and upper bound are collected in **UB** = $\{5, 5\}$. The length criterion, **CRITERION** = 1, of sub-interval for every variable and the numbers of sub-intervals for the first and second variables are

$$nDivide[0] = \frac{UB[0] - LB[0]}{CRITERION} = 10, \quad (1)$$

$$nDivide[1] = \frac{UB[1] - LB[1]}{CRITERION} = 10, \quad (2)$$

So the number of the total sub-domains is

$$nTotal = nDivide[0] \times nDivide[1] = 100. \quad (3)$$

Obviously, this model is very easy to be extended into high dimensional problems.

Through the above example we will show the distributional uniformity of the initial solutions in the definition domain generated by the I&I model. The definition domain is divided into 100 sub-domains and the length and the width are both 1 of every sub-domain. **Figure 1** shows the distribution of 100 random initial solutions independently generated from the total definition domain for **Six-Hump Camel-Back Function**. There are thirty-four sub-domains which are gray in **Figure 1**

from where no initial solutions are generated. On the contrary, in some other sub-domains there are two, three, even four initial solutions. That is to say, the initial solutions which are randomly generated do not uniformly distribute in the definition domain. **Figure 2** displays another situation of the initial solutions generated by the I&I model that they uniformly distribute in 100 sub-domains. It is easily observed that the I&I model is more possible to explore and exploit the optimal solution of a given problem.

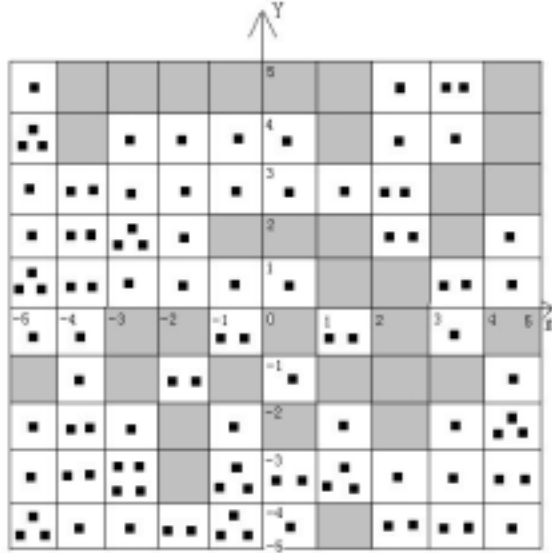


Figure 1 Random Distribution of 100 Independent Initial Solutions of **Six-Hump Camel-Back Function** in the Definition Blocks

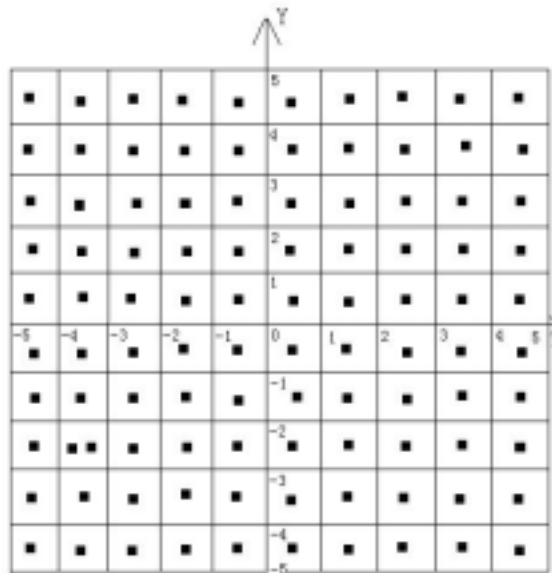


Figure 2 Uniform Distribution of 100 Initial Solutions of **Six-Hump Camel-Back Function** in the Definition Blocks for I&I Model

The procedure of randomly generating initial points in every sub-domain and every hill-climbing procedure are as follows:

Procedure of Initialization

```
for (i= 0; i<nDivide[0]; i++)
for (j= 0; j<nDivide[1]; j++)
generate an initial point at sub-domain D(i, j).
```

Procedure of Evaluation

```
for (i= 0; i<nDivide[0]; i++)
for (j= 0; j<nDivide[1]; j++)
hill-climb from the initial point of D(i, j) and obtain an
optimum O(i, j) of D(i, j).
Return the best results from all the O(i, j) (i= 1, . . . ,
nDivide[0], j= 1, . . . , nDivide[1]).
```

3. BENCHMARK FUNCTIONS

Eight benchmark functions [12, 13] were used in our experimental studies to show the effectiveness of our isolated island model. These benchmarks are widely used in comparison among the stochastic search algorithms.

Bohachevsky Function #1:

$$f_1 = x_1^2 + 2x_2^2 - 0.3 \cos(3x_1) - 0.4 \cos(4x_2) + 0.7,$$

$$-50 \leq x_i \leq 50, \min(f_1) = f_1(0, 0) = 0.$$

Bohachevsky Function #2:

$$f_2 = x_1^2 + 2x_2^2 - 0.3 \cos(3x_1) \cos(4x_2) + 0.3,$$

$$-50 \leq x_i \leq 50, \min(f_2) = f_2(0, 0) = 0.$$

Schaffer Function:

$$f_3 = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2},$$

$$-100 \leq x_i \leq 100, \min(f_3) = f_3(0, 0) = 0.$$

Easom Function:

$$f_4 = -\cos(x_1) \cos(x_2) \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2),$$

$$-100 \leq x_i \leq 100, \min(f_4) = f_4(\pi, \pi) = -1$$

Shubert Function:

$$f_5 = \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \times \sum_{i=1}^5 i \cos[(i+1)x_2 + i]$$

$$-10 \leq x_i \leq 10, \min(f_5) = -186.73.$$

Rosenbrock Function:

$$f_6 = -100(x_1^2 - x_2)^2 - (1 - x_1)^2,$$

$$-2.048 \leq x_i \leq 2.048, \min(f_6) = -3905.9262.$$

Six-Hump Camel-Back Function:

$$f_7 = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2,$$

$$-5 \leq x_i \leq 5, \min(f_7) = f_7(0.08933, -0.7126) =$$

$$f_7(-0.08983, 0.7126) = -1.0316285.$$

Goldstwin-Price Function:

$$f_8 = [1 + (x_1 + x_2 + 1)^2 \times$$

$$(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$$

$$[30 + (2x_1 - 3x_2)^2 \times$$

$$(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)],$$

$$-2 \leq x_i \leq 2, \min(f_8) = f_8(0, -1) = 3.$$

4. EXPERIMENTAL STUDIES

In the experiments, a precision of six digits after the decimal point (**PRECISION** = 6) is supposed and there are three important parameters for hill-climbing method and isolated island model: **CRITERION**, **DIFF** and **LOOP**. **CRITERION** is to balance the length of the sub-intervals and the total length of the definition interval for each variable. The value of **CRITERION** is also a trade-off between computational effectiveness and costs. **DIFF** is the largest perturbation distance for every local search. That is to say, the micro perturbation at the initial point is decided by the following formula, where **ZERO** = $10^{-\text{PRECISION}}$ and $\text{rand}(0, 1)$ is a random decimal fraction between 0 and 1.

$$\text{delta} = \text{ZERO} + (\text{DIFF} - \text{ZERO}) \times \text{rand}(0, 1) \quad (4)$$

LOOP is the repetition number of hill-climb at every sub-domain from the initial point.

The local perturbation we adopted is as the following steps. The (s4) and (s5) ensure the perturbation not to exceed the definition domain, at the same time, they are especially effective for the optimization problems whose global optima lie at their definition boundaries (such as f_6).

- (s0) Evaluate function value before perturbation, f_0 ;
- (s1) Generate a micro distance **delta** use Eq.(4);
- (s2) Generate a perturbation direction **direction** = 1 or -1 with 50% probability respectively;
- (s3) Perturb the j th component of variable X:
 $x_j = x_j + \text{direction} \times \text{delta}$;
- (s4) if ($x_j < \text{lower bound of } j\text{th component of variable X}$)
 {
 $x_j = \text{lower bound or } x_j - 1.5 \times \text{direction} \times \text{delta}$ with 50% probability respectively;
 }
- (s5) if ($x_j > \text{upper bound of } j\text{th component of variable X}$)
 {
 $x_j = \text{upper bound or } x_j - 1.5 \times \text{direction} \times \text{delta}$ with 50% probability respectively;
 }
- (s6) Evaluate function value after perturbation, f_1 ;
- (s7) if ($f_1 > f_0$)
 accept the micro perturbation;
 else
 resume j th component of variable X.

For the sake of impartiality, we set the number of hill-climb for the local search method as the following formula.

$$\text{nClimb} = \text{nDivide}[0] \times \text{nDivide}[1] \times \text{LOOP} \quad (5)$$

Generally speaking three parameters vary with the length of the variable intervals. But we just choose two values for these parameters according to the length of the variable intervals to avoid fine tuning. The experimental results are found in **Table 1** and **Table 2** which are based on 50 independent trials for every benchmark. The (*C*, *D*, *L*) stands for the values of (**CRITERION**, **DIFF**, **LOOP**) of the isolated island model. The *nOptima* shows the number of reaching the optima for two algorithms in 50 trials. *Average* and *Worst* are the average and worst results in 50 independent trials. Program is written by C language.

From **Table 1**, conclusions that I&I model predominating local search and lower **CRITERION** outperforming larger

CRITERION for I&I model are easily reached. Especially important, for multimodal functions f_5, f_7 , the respective occurrence times for different solutions being found have **no statical difference** for I&I algorithm. Function f_3 has an infinite number of subminima which enclose the global minimum, 0, with a distance about 3.14. However, we adopt **CRITERION** = 5 in our experiments. So the global minimum and some sub-minima of it are certain to locate in the same sub-domain and the searching engine is easy to be trapped by sub-minima. We do another group experiments with **CRITERION** = 1, **DIFF** = 0.02 and other parameters remain unchanged. Isolated island model all found the global optimum in 20 trials.

Obviously, this model is very easy to be extended into high dimension optimization problems, though the experiments are based on the two dimensional problems.

5. POSSIBLE PARALLEL STRATEGIES FOR THE I&I MODEL

In order to prove the effects of I&I model, a simple hill-climbing strategy is used in our isolated island algorithm, however, highly satisfiable results are obtained from our experiments. Some even better strategies can be incorporated into the present isolated island model, such as genetic algorithm [15, 16, 17], simulated annealing [1, 18] and tabu search [19]. If the total definition domain is divided into several exclusive sub-domains and a sub-population is independently processed on a sub-domain, we believe that some even better results can be reported than **DGA** does. Every sub-population in **DGA** all the same searches the solutions in the total definition domain, however, sub-population in the isolated island model is just processed in a much smaller sub-domain. At least, the initial solutions of genetic algorithms can be generated based on our proposed isolated island model, which ensures that the initial solutions can be distributed in every definition blocks.

The simulated annealing strategy is further discussed when it is incorporated into the I&I model. It is obvious that the results of simple hill-climbing strategy being incorporated into I&I model are very good except for f_1, f_3 and f_4 functions which can be seen from **Table 1**. So the I&I model will only consider them when simulated annealing is incorporated into to replace the simple hill-climbing strategy. The constant initial temperature

$$T_0 = 10 \times \text{LOOP} = 5000$$

and the **geometric annealing schedule** [1]

$$T = T \times \alpha, \text{ where } \alpha = 0.92 \quad (6)$$

are adopted to anneal.

The column **C** in **Table 2** is the value of parameter

Table 2 Performance Comparison between Simulated Annealing and Hill-climbing Strategy for I&I Model

Fun	Simulated Annealing		Hill Climbing	
	C	nOptima	C	nOptima
f_1	5	45	5	34
f_4	5	35	5	24
f_3	5	16	5	15
f_3	3	34		

Table 1 Performance Comparison for I&I Model and Local Search Method

Fun	(C, D, L)	I&I Model			Local Search		
		nOptima	Average	Worst	nOptima	Average	Worst
f1	(5, 0.4, 500)	34	0.132137	0.412933	10	0.330341	0.412926
f2	(5, 0.4, 500)	49	0.008258	0.412927	13	0.196582	0.412927
f3	(5, 0.4, 500)	15	0.006801	0.009716	3	0.017385	0.037224
f4	(5, 0.4, 500)	24	-0.540035	-0.000081	0	0 0.000000	0 0.000000
f5	(1,0.02,500)	50	-186.730908	-186.730902	2	-34.833746	-7.722748
f6	(1,0.02,500)	50	-3905.926227	-3905.926227	24	-3206.206111	-420.432847
f7	(1,0.02,500)	50	-1.0316284	-1.0316281	12	-0.086583	2.104250
f8	(1,0.02,500)	50	3	3	15	172.02000	840.000000

CRITERION used in the algorithms. The *nOptima* is the number of finding the optima in 50 independent trials. The performance of simulated annealing with I&I model is much better than the simple hill-climbing strategy for function *f1* and *f4*. The ratio of finding optima for *f1* and *f4* are increased by 20 percent or so, but the performance of function *f3* has no significant improvement between simulated annealing and hill-climbing strategy for I&I model when the parameter **C** is 5. The reason may be the fact that an infinite number of sub-minima of *f3* enclosing the global minimum with a distance about 3.14. However, we can obtain much encouraging result with *nOptima* = 34 if we adjust **CRITERION** to be 3 from 5, which is just a little smaller than 3.14. Then the ratio of finding the optima of *f3* is increased by 40 percent or so. In a word, I&I model with simulated annealing outperforms I&I model with hill-climbing strategy.

6. CONCLUSION

An isolated island (I&I) distributed computation model is proposed in this paper based on the hill-climbing method. Experiments based on some classical benchmarks show the superiority of this model. One better strategy, simulated annealing, is also discussed when it is incorporated into the proposed I&I model for some benchmarks and obtain even better results. Experiments have proved the dominance of the isolated island model. Especially important, for multimodal functions, the respective occurrence times for different solutions being found have **no statical difference** for I&I distributed model.

Based on the previous distributed computing model and our I&I model, two stages distributed computing model will be discussed in the future work. We hope such work will boost the idea of the isolated island distributed model and even better parallel computational algorithms are proposed.

7. ACKNOWLEDGEMENT

I would like to thank my teacher, Xiao-shan Gao, Institute of Systems Science of Academy of Chinese Sciences and the suggestions of DCABES committee.

8. REFERENCES

- [1] L. S. Kang, Y. Xie, S. Y. You and Z. H. Luo, Non-Numrical Parallel Computation (1st Vol.) Simulated Annealing (in Chinese), Science Press, Beijing, China, 2000.
- [2] T. Ray, K.M. Liew, Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior, IEEE Trans. Evol. Comput, Vol7, No.4, 2003.
- [3] H. Muhlenbein, M. Schomisch and J. Born, The Parallel Genetic Algorithm as Function Optimizer, 4th Int. Conf. Genetic Algorithms, R. Belew and L. B. Booker, Ed. San Mateo, CA: Morgan Kaufmann, p271-278, 1991.
- [4] F. Herrera, M. Lozano, Gradual Distributed Real-Coded Genetic Algorithms, IEEE Trans. Evol. Comput, Vol4, No.1, 2000.
- [5] J.C. Potts, T.D. Giddens and S.B. Yadav, The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection, IEEE Trans. Syst., Man and Cybern., Vol24, No.1, 1994.
- [6] T. Kuo, S.Y. Huwang, A Genetic Algorithm with Disruptive Selection, IEEE Trans. Syst., Man and Cybern., Vol26, No.2, 1996.
- [7] S.W. Mahfoud, Niching Methods for Genetic Algorithm, Univ. Illinois at Urbana Champaign, Illinois Genetic Algorithms Lab., IlliGAL Rep. 95001, 1995.
- [8] J.D. Schaffer, R.A. Caruana, L.J. Eshelman and R. Das, A Study of Control Parameters Affecting Online Performance of Genetic Algorithms, Proc. 3rd Int. Conf. Genetic Algorithms, p51-60, 1989.
- [9] P. Adamidis, V. Petridis, Co-operating Populations with Different Evolution Behaviours, Proc. 3rd IEEE Conf. Evol. Comput., New York: IEEE Press, p188-191, 1996.
- [10] R. Tanese, Distributed Genetic Algorithms, Proc. 3rd Int. Conf. Genetic Algorithms, J.D. Schaffer, Ed. San Mateo, CA, p434-439, 1989.
- [11] Y. X. Li, X. F. Zou, L. S. Kang and Z. Michalewicz, A New Dynamical Evolutionary Algorithm Based on Statistical Mechanics, Journal of Computer Science and Technology, 18, 3, 361-368, 2003.
- [12] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, (3rd Edition), Springer, 1996.
- [13] X. Yao, Y. Liu and G. M. Lin, Evolutionary

- Programming Made Faster, IEEE Trans. Evol. Comput, Vol3, No.2, 1999.
- [14] D. H. Wolpert, W. G. Macready, No Free Lunch Theorems for Optimization, IEEE Trans. Evol. Comput, Vol1, Apr. 1997.
- [15] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA: Addison-Wesley, 1989.
- [16] G. L. Chen, X. F. Wang, Z. Q. Zhuang and D. S. Wang, Genetic Algorithms and Applications (in Chinese), posts&telecom press, China, 1996.
- [17] M. Zhou, S. D. Sun, Genetic Algorithms: The ory and Applications (in Chinese), National University of Defense Technology Press, China, 1999
- [18] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by Simulated Annealing, Science, 220, p671-680, 1983.
- [19] F. Glover, Tabu Search - Part I, ORSA J. Computing 1: p190-206, 1989.



Xinchao Zhao is a PhD Candidate of key laboratory of mathematics mechanization, Academy of Mathematics and Systems Science, Chinese Academy of Sciences. His research interests include evolutionary computation, global optimization and bioinformatics.

Algorithm for Solving the Symmetric Five-Diagonal Toeplitz Linear Equations*

Ran Ruisheng¹, Huang Tingzhu²

¹ School of Computer Sci. and Eng., ² School of Appl. Math.,
University of Electronic Science and Technology of China, Chengdu, sichuan, P. R. China
E-mail: rshran@163.com or tzhuang@uestc.edu.cn Tel: (028) 83202637

ABSTRACT

The symmetric five-diagonal Toeplitz system appears in many problems came from mathematics and applied science. In this paper, a method for solving the system is presented. The derivation of the result is principally based on a matrix split and a tridiagonal Toeplitz factorization of a symmetric five-diagonal Toeplitz matrix modified. The symmetric circulant five-diagonal Toeplitz system is also considered.

Keywords: The symmetric five-diagonal Toeplitz linear equation; Matrix split; Perturbation; Algorithm

1. INTRODUCTION

Five-diagonal matrices often arise in several fields such as numerical solution of ordinary and partial differential equation, interpolation problems, boundary value problems, etc. In the solution of one-dimensional elliptic equation subject to periodic boundary conditions and the approximation of periodic function using splines, banded circulant matrices are also often encountered.

In this paper, our interest is focused on solving the symmetric five-diagonal Toeplitz system and the symmetric five-diagonal circulant Toeplitz system. For the recent years, many methods have been proposed for solving the systems, e.g. [1~5], etc. These methods have used the perturbation of the systems with appropriate matrix split. They are competitive with Gaussian elimination in term of arithmetic operations and storage requirements.

In [4], for a five-diagonal Toeplitz matrix modified in the first and the last element of the principal diagonal, it has been decomposed as the product of two tridiagonal Toeplitz matrices. With the factorization at hand, the solution of the system required smaller operation than the method based on the LU factorization of the five-diagonal Toeplitz matrix modified.

In this paper, we consider the symmetric five-diagonal Toeplitz linear equations given by

$$Mx = b, \quad (1)$$

where

$$M = \begin{pmatrix} a & b & c & & & \\ b & a & b & & & \\ c & b & a & & & \\ & . & . & . & . & \\ & & . & . & . & \\ & & & c & b & a & b \\ & & & & c & b & a \end{pmatrix}$$

and we assume that M is strictly diagonally dominant. Based on the factorization in [4], we proposed a new algorithm for solving the Eq. (1) in this paper. Firstly, we split the matrix M as the addition of a five-diagonal Toeplitz matrix modified M_p and a correction matrix M' ; then we decompose the matrix M_p as the product of two tridiagonal Toeplitz matrices; finally, with the help of Woodbury's formula, an algorithm for solving the Eq. (1) is established.

2. SPLITTING THE FIVE-DIAGONAL TOEPLITZ MATRIX

Generally, one attempts to give the Toeplitz LU factorization of the symmetric five-diagonal Toeplitz matrix M . However there are no Toeplitz matrix L and U such that $M=LU$, and he may fail. In [3], after modifying the matrix M with the appropriate perturbation, the Toeplitz LU factorization has been presented.

Here, we split the matrix M into

$$M = M_p + M',$$

where

$$M_p = \begin{pmatrix} a-\alpha & b & c & & & \\ b & a & b & & & \\ c & b & a & & & \\ & . & . & . & . & \\ & & . & . & . & \\ & & & c & b & a & b \\ & & & & c & b & a-\beta \end{pmatrix}, \text{ and}$$

$$M' = \begin{pmatrix} \alpha & & & & & \\ & 0 & & 0 & & \\ & & 0 & & & \\ & & & 0 & & \\ & 0 & & & 0 & \\ & & & & & \beta \end{pmatrix}.$$

The matrix M_p is a symmetric five-diagonal Toeplitz matrix modified in the first and the last principal diagonal. For the non-symmetric case, the matrix is decomposed as the product of two Toeplitz tridiagonal matrices in [4]. Here, we have $M_p = PP^T$. Where, P is a tridiagonal Toeplitz matrix and denoted

$$\text{by } P = \text{trid}_n[x, y, z], \text{ i.e., } P = \begin{pmatrix} x & y & & & \\ z & x & y & & \\ & 0 & 0 & 0 & \\ & & z & x & y \\ & & & z & x \end{pmatrix},$$

and the matrix P^T is the transpose of P .

* This work is supported by NSFC (60372012).

It is easy to verify that the factorization exists if and only if the following equalities are satisfied:

$$\begin{aligned}x^2 + y^2 + z^2 &= a, \\xz + yx &= b, \\yz &= c, \\z^2 &= \alpha, \\y^2 &= \beta.\end{aligned}$$

The entries of X, Y, Z may be determined by the method introduced in [4]. Here we do not state the method and only apply it.

3. SYMMETRIC FIVE-DIAGONAL TOEPLITZ LINEAR EQUATIONS

Firstly, let us consider the solution of the linear equations

$$M_p x = q. \quad (2)$$

Where q is an any given n dimension vector.

In fact, with the factorization of $M_p = PP^T$, Eq. (2) is equivalent to the linear equations of

$$\begin{aligned}Py &= q \\P^T x &= y\end{aligned} \quad (3)$$

Where, the two equations are tridiagonal Toeplitz equations and may be solved by a suitable method.

Here, we use the regular LU factorization method for both tridiagonal Toeplitz equations. In fact, only one LU factorization of P is necessary because the matrix P^T is the transpose of P . Let the matrix P is decomposed as $P = LU$. Then Eq. (3) is equivalent to the equations

$$\begin{aligned}Lr &= p, \quad Uy = r; \\U^T s &= y, \quad L^T x = s.\end{aligned} \quad (4)$$

However, if P is strictly diagonal dominant, i.e.,

$$|x| > |y| + |z|,$$

then P can be written

$$P = P' + E,$$

where

$$P' = \begin{pmatrix} x-\delta & y & & & \\ & z & x & y & \\ & & O & O & O \\ & & & z & x & y \\ & & & & z & x \end{pmatrix}, \quad E = \begin{pmatrix} \delta & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \end{pmatrix}.$$

Furthermore, P^T can be expressed as the product of two Toeplitz matrix L and U , where

$$L' = \begin{pmatrix} d & & & & \\ \gamma & d & & & \\ & O & O & & \\ & & & \gamma & d \end{pmatrix}, \quad U' = \begin{pmatrix} \tau & \sigma & & & \\ & O & O & & \\ & & O & \sigma & \\ & & & O & \sigma \\ & & & & \tau \end{pmatrix}.$$

Then the equations $Py=q$ may be solved by the perturbed equations $Py = P'y + Ey$. Note that $P = LU'$, the solution of the equations $P'y = q'$ is quite easy and costs a few arithmetic operations and storage requirements. In fact, the above method of solving the equations $Py=q$ has been proposed in [3][7].

Now, let us consider the symmetric five-diagonal Toeplitz

system (1). For the split of M in section 2

$$M = M_p + M'.$$

Viewing $M = M_p + M'$ as

$$M = M_p + M'^T,$$

and by the Woodbury's formula [6], we have

$$M^{-1} = M_p^{-1} - M_p^{-1} M' (I + M_p^{-1} M')^{-1} M_p^{-1}, \quad (5)$$

where I is a identity matrix, I^T is the transpose of I .

Let x denotes the solution of Eq. (1) that means $x = M^{-1}b$. Then by Eq. (5), we get

$$x = M_p^{-1}b - M_p^{-1}M'(I + M_p^{-1}M')^{-1}M_p^{-1}b \quad (6)$$

It is clear that x can be derived from Eq. (5). The calculation of Eq. (5) is principally the calculation of the expressions:

$$M_p^{-1}b, \quad M_p^{-1}M', \quad (I + M_p^{-1}M')^{-1}.$$

Note that the special structure of the correction matrix M' , it is clear that the calculation of $M_p^{-1}M'$ is equivalent to the calculation of the vectors

$$\alpha M_p^{-1}e_1, \quad \beta M_p^{-1}e_n.$$

Thus it is necessary to know the entries of the vectors $M_p^{-1}b$, $M_p^{-1}e_1$ and $M_p^{-1}e_n$.

With Eq. (3), the above vectors can be derived by replacing the vector q with the vector b , e_1 and e_n . For the convenience, denote the vectors $M_p^{-1}b$, $M_p^{-1}e_1$ and $M_p^{-1}e_n$ by the vectors $x^p = (x_i^p)$, $w = (w_i)$ and $v = (v_i)$ respectively.

After obtaining the matrix $M_p^{-1}M'$, the calculation of $(I + M_p^{-1}M')^{-1}$ is not difficult. Due to the special structure of the matrix $M_p^{-1}M'$, the matrix $M_p^{-1}M'(I + M_p^{-1}M')^{-1}$ has also special structure similar to $M_p^{-1}M'$. Here, we omit the details of deducing the matrix

$$M_p^{-1}M'(I + M_p^{-1}M')^{-1}$$

and give the following results directly.

We get the solution of Eq. (3)

$$\begin{aligned}x_i &= x_i^p - t \{ [w_i(1 + v_n) - v_i w_n] x_i^p + \\ & [v_i(1 + w_1) - w_i v_1] x_n^p \} \quad (i = 1, 2, \Lambda, n),\end{aligned}$$

where

$$t = \frac{1}{1 + w_1 + v_n - w_1 v_n - w_n v_1}.$$

If let

$$g_i = w_i(1 + v_n) - v_i w_n,$$

$$h_i = v_i(1 + w_1) - w_i v_1,$$

the above expression may be reformulated as

$$x_i = x_i^p - t(g_i x_i^p + h_i x_n^p) \quad (i = 1, 2, \Lambda, n) \quad (7)$$

Then an algorithm for solving the symmetric five-diagonal Toeplitz system has been established. We summarize it as follows.

Algorithm 1.

Step 1. Decompose M_p as $M_p = PP^T$;

Step 2. Decompose P as LU ;

Step 3. Replace the vector q with the vector b , e_1 and e_n in Eq. (4) and solve the system, denote the solution by $x^p = (x_i^p)$, $w = (w_i)$ and $v = (v_i)$ respectively;

Step 4. Compute the entries of t , g_i and h_i ;

Step 5. Compute the solution of Eq. (1) by Eq. (7).

4. SYMMETRIC CIRCULANT FIVE-DIAGONAL TOEPLITZ LINEAR EQUATIONS

In this section, we consider the symmetric five-diagonal circulant Toeplitz linear equations order of n given by

$$Nx = b, \quad (8)$$

where

$$N = \begin{pmatrix} a & b & c & & & c & b \\ b & a & b & & & & c \\ c & b & a & & & 0 & \\ & . & . & . & . & & \\ & & . & . & . & . & \\ & & & . & . & . & \\ 0 & & & & . & . & . \\ c & & & & & c & b & a & b \\ b & c & & & & c & b & a & b \end{pmatrix}$$

$$= N(a, b, c, 0, \Lambda, 0, c, b)$$

and we assume that N is strictly diagonally dominant.

For the case of symmetric circulant five-diagonal Toeplitz coefficient matrix, the linear equations have been investigated in [3]. Here, we combine the method introduced in [3] with the results obtained in section 3.

Let

$$\begin{aligned} \hat{b} &= (b_1, b_2, \Lambda, b_{n-2})^T, \\ \tilde{b} &= (b_{n-1}, b_n)^T, \quad b = (\hat{b}, \tilde{b})^T, \\ \hat{x} &= (x_1, x_2, \Lambda, x_{n-2})^T, \\ \tilde{x} &= (x_{n-1}, x_n)^T, \quad x = (\hat{x}, \tilde{x})^T, \\ Q &= \begin{pmatrix} c & b \\ 0 & c \end{pmatrix}, \\ V^T &= (Q^T, 0^T, Q), \\ A &= \begin{pmatrix} a & b \\ b & a \end{pmatrix}, \end{aligned}$$

where 0 is $(n-6) \times 2$ zero matrix.

So we can divide Eq. (8) as follows

$$\begin{pmatrix} M & V \\ V^T & A \end{pmatrix} \begin{pmatrix} \hat{x} \\ \tilde{x} \end{pmatrix} = \begin{pmatrix} \hat{b} \\ \tilde{b} \end{pmatrix}.$$

In [3], the solution of Eq. (8) has been expressed as

$$\begin{aligned} \hat{x} &= M^{-1}r + M^{-1}V(A + V^T M^{-1}V)^{-1}V^T M^{-1}r, \\ \tilde{x} &= A^{-1}(\tilde{b} - V^T \hat{x}), \end{aligned} \quad (9)$$

where $r = \hat{b} - VA^{-1}\tilde{b}$.

For Eq. (9), it is principal to compute

$$M^{-1}r, M^{-1}V.$$

And $M^{-1}V$ may be divided as $M^{-1}q_1$ and $M^{-1}q_2$, where

$$q_1 = (c, 0, \Lambda, 0, c, b), \quad q_2 = (b, c, 0, \Lambda, 0, c).$$

Note that M is a symmetric five-diagonal Toeplitz matrix, putting $b = r$, $b = q_1$ and $b = q_2$ in Eq. (1) respectively, these vector $M^{-1}r$, $M^{-1}q_1$ and $M^{-1}q_2$ may be computed by the algorithm presented in Section 3.

5. CONCLUSION

Let us consider the operations for the method presented in this

paper. In the algorithm 1, the cost of step 1 is $O(1)$, as is known from [4]; with the regular LU factorization method solving the tridiagonal linear equations, the cost is $8n$ while the cost is $5n+2t$ if we apply the method taken from [7]. So, $30n$ or less operation may be required in step 2 and 3; it is easy to know that the cost is $6n$ in step 4 and the cost is $4n$ in step 5. So, the realization of the algorithm for solving the five-diagonal Toeplitz system needs $40n$ or less operation.

The method presented in this paper is very stable one. For the circulant case, our method is also competitive with the other method [8] for solving the circulant linear system.

6. REFERENCES

- [1] L.E.Garey, R.E.Shaw, "A parallel algorithm for solving Toeplitz linear systems", Appl. Math. and Comp., 100, 1999, pp.241-247.
- [2] L.E.Garey, C.J.Gladwin and R.E.Shaw, "An approximate solution for a system with a symmetric matrix", Comp. Math. Appl., Vol.34, No.12, 1997, pp. 25-31.
- [3] S.M.El-Sayed, I.G. Ivanov and M.G. Petkov, "A new modification of the Rojo method for solving symmetric circulant five-diagonal systems of linear equations", Computers Math. Appl., Vol.35, No.10, 1998, pp. 35-44.
- [4] F. Diele, L.Lopez, "The use of factorization of five-diagonal matrices by tridiagonal Toeplitz matrices", Appl. Math. Lett., Vol.11, No.3, 1998, pp. 61-69.
- [5] S.S.Nemani, L.E.Garey, "Parallel algorithm for solving tridiagonal and near-circulant systems", Appl.Math. and Comp., 130, 2002, pp.285-294.
- [6] G.H. Golub, C.F. Van Loan, Matrix Computation, third ed., Beijing: the Science Press, 2001.
- [7] W.M. Yan, K.L. Chung, "A fast algorithm for solving special tridiagonal systems", Computing, 52,1994, pp.203-211.
- [8] M. Chen, "On the solution of circulant linear systems", SIAM J. Numer. Anal. 24,1987, pp.668-683.



Ran Ruisheng graduated from School of Applied mathematics with Master and is now pursuing Ph.D. at School of Computer Sci. and Eng., University of Electronic Science and Technology of China. He has published several journal papers. His research interest is principally scientific computing.



Huang Tingzhu is a Professor, Dr. and the Dean of School of Applied Math of University of Electronic Science and Technology of China. He graduated from Dept of Math, Xi'an Jiaotong University in 2001 with Ph.D. He was a Visiting Professor of Appl. Math Institute of Chinese Science Academia (1999), a Visiting Professor of Hong Kong Baptist University (2002). He has

published several books and over 80 journal papers. His research interests are scientific computing, numerical algebra, and matrix analysis with applications etc.

The Fuzzy Inference Based on Genetic Algorithm

Zhang Jianhua, Jiang Qian

School of Information and Science & Engineering, Shenyang University of Technology,
Shenyang, Liaoning Province, 110023, P. R. China
E-mail: zhangjianhua2002@hotmail.com

ABSTRACT

This article analyzes the fuzzy conversion and fuzzy inference in the fuzzy mathematical theory. It has pointed out that the introduction of genetic algorithm, which may confer the ability of inference knowledge on Fuzzy control, Pattern Recognition, Modeling and simulation of complex systems, Identification and estimation.

Keywords: genetic algorithm, fuzzy algorithm

1. INTRODUCTION

In recent years, the Computer Vision and the Control technology have been accelerated with that Intelligent Algorithm and Soft Computing were introduced into the research fields. The intelligent algorithms usually refer to the Genetic Algorithm, Taboo Search, Simulated Annealing, Artificial Neural Network, etc.

As, Pattern recognition system, the processing of data obtaining include various information, such as words, pictures, image, sounds, are often measured, sampled and quantified. At the same time, the logical relations, including the fuzzy logical relations, between physical parameters should be considered and all of them must be input to the computers. In the processing of characteristic extraction and choice, we can take the procedure of classified determination only after the conversion in general. In the procedure of classified determination, what we should care more about is which algorithm is more suitable (quickly and efficient) to adopt when we analyze the problem, so that we are able to get the relevant conclusions. In the above procedures, we may, we have to employ the intelligent algorithms and the soft computing to solve the problems.

2. THE FUZZY CONVERSION AND FUZZY INFERENCE

The fuzzy conversion based on fuzzy relations is a very important operating procedure in the fuzzy computing. We can acquire the required output fuzzy quanta from the input ones by fuzzy conversion.

In linear algebra, matrix $A=(a_{ij})_{m \times n}$ and series of vector, X have been given, then $Y=AX$ can be obtained, in which the element of the vector Y may be computed according to the following formula.

$$y_i = \sum_{j=1}^n a_{ij}x_j, \quad i=1,2,\dots,m \quad (1)$$

In the fuzzy situation, A and B are the fuzzy subsets in the fuzzy sets X and Y , a fuzzy matrix is given.

$$R = F(X \times Y), \quad R=[r_{ij}], \quad 0 \leq r_{ij} \leq 1 \quad (2)$$

$$A=(x_1, x_2, \dots, x_n), \quad 0 \leq x_i \leq 1$$

Thus, we have $A \circ R = B$

This formula is called fuzzy conversion. Namely, the fuzzy subset B may be procured by the composition operation,

$$B = A \circ R$$

If the fuzzy conversion is applied in the control system, R in the above relations can denote the dynamic relations between the A (input) and the B (output). That is, if R denotes some logical causality and is input into A , the synthetic judgment result B is obtained from fuzzy conversion. We also call this kind of process in which the result is acquired from fuzzy conversion the fuzzy inference.

The fuzzy inference is named fuzzy logic inference as well. It refers to the process in which a new fuzzy proposition is given. Thus, it is a kind of approximate inference. Composition Rule of Inference (CRI), L.A.Zadeh presented⁰ is a sort of fuzzy inference algorithm that is applied quite widely. According to the formula (1), we can describe it further.

Given: A is the fuzzy set on X , B is the fuzzy set on Y , R denotes their fuzzy containing relations, then

$$R = A \times B = (A \times B) \cap (A^c \times Y) \quad (3)$$

Take the membership function:

$$\begin{aligned} \mu_R(x,y) &= \mu_A(x) \wedge \mu_B(y) \\ &= (1 - \mu_A(x)) \wedge (\mu_A(x) \wedge \mu_B(y)) \end{aligned} \quad (4)$$

This is,

$$R(x,y) = [A(x) \wedge B(y)] \wedge [1 - A(x)] \quad (5)$$

In addition, we have the following inference rule FMP (Fuzzy Modus Ponens):

If $R = A \times B$, for the given A^* , $A^* \in X$,

we can get the conclusion B^* , $B^* \in Y$,

B^* is shown in the following formula:

$$B^* = A^* \circ R$$

$$B^*(y) = A^*(x) \circ R(x,y)$$

$$= \sup_{x \in X} \{A^*(x) \wedge [A(x) \wedge B(y)] \wedge [1 - A(x)]\} \quad (6)$$

We still have the following inference rule FMT (Fuzzy Modus Tollens):

If $R = A \times B$, for the given B^* , $B^* \in Y$,

we can get the conclusion A^* , $A^* \in X$,

A^* is shown in the following formula:

$$A^* = R \circ B^*$$

$$A^*(x) = R(x,y) \circ B^*(y)$$

$$= \sup_{x \in X} \{[A(x) \wedge B^*(y)] \wedge [1 - A(x)] \wedge B^*(y)\} \quad (7)$$

Another conclusion that we must know as follows

IF A \rightarrow F(X), B \rightarrow F(Y), C \rightarrow F(Z), when A* and B* are known,
There is the expression as

$$R = A \times B \times C = (A \times B) \times C$$

Or, there is the expression as

$$R(x, y, z) = A(x) \times B(y) \times C(z)$$

That is

$$\begin{aligned} \text{If A and B, Then } C^* &= (A^* \times B^*) \times R \\ A^* &\rightarrow F(x), B^* \rightarrow F(y), C^* \rightarrow F(z) \end{aligned} \quad (8)$$

3. FUZZY SCHEMA RECOGNITION METHODS

There are three kinds of schema recognition methods based on fuzzy set theory as follows.

Membership Principle Recognition Method

Given: There are n fuzzy subsets A_1, A_2, \dots, A_n in the set U, and every A_i has its membership function $\mu_{A_i}(x)$,

such as $x_0 \in U$,

$$\text{when: } \mu_{A_i}(x_0) = \max_{x \in U} [\mu_{A_1}(x_0), \mu_{A_2}(x_0), \dots, \mu_{A_n}(x_0)] \\ \Rightarrow x_0 \in A_i$$

Namely, we can conclude that x_0 belongs to A_i

Neighbor-choosing Principle Recognition Method

In the realistic schema recognition, the recognized target is usually a subset of U (but not a element of U), what we will discuss is how close the two fuzzy subsets are.

Where A and B are two fuzzy subsets on U.

The definition of the closeness between the two subsets is:

$$(A, B) = [A \cdot B + (1 - A \cdot B)] / 2$$

$$\text{in which, } A \cdot B = \max_{x \in U} (A(x) \cdot B(x))$$

$$A \cdot B = \max_{x \in U} (A(x) \cdot B(x))$$

Fuzzy Cluster Analysis Method

In the schema recognition, sometimes we do not know how much kind of we should classify before the classification. We can only put the similar schema together according to certain fuzziness. This classified method is called fuzzy cluster analysis method.

4. INTRODUCING THE FUZZY INFERENCE BASED ON GENETIC ALGORITHM

Fuzzy control is based on the Fuzzy inference. On the basis of the physical characteristics of the controlled system, it is a kind of intelligent control of simulating the thought and experience of human being. Because it depends on the fixed inference rules to control, it is for lack of self-taught and self-adapted abilities. We can introduce the genetic algorithm to make fuzzy inference rules change according to real situations. Consequently, it confers the ability to acquire inference knowledge automatically on the fuzzy control processor.

The Structure and Feature of the Genetic Algorithm

The genetic algorithm is the search algorithm based on natural selection and hereditary principle of genes. It was different from the traditional search algorithm. First of all, the genetic algorithm starts to search from a group of initial solutions

(also called population) random. Every individual in the population is a code string, called individual string (or called chromosome), it is one of the solution of the problem. Chromosomes evolve continually from one generation to another, and are called heredity. The born chromosomes of the next generation are called offspring, which are formed by genetic operations (namely, crossover and mutation operations). In the formation of the new generation, fitness is used to measure goodness and badness of chromosomes. According to the size of fitness, a part of offspring is selected and another part of offspring is eliminated. Therefore, the size of the population is kept constant. The higher fitness of chromosomes is, the higher probability of selection is. After some generations, the chromosomes were produced, and the operation must be stop. It is likely to be the optimal or sub-optimal solution.

Converting the Fuzzy Inference Pattern

There is an illustration as follows.

Given, the inference rule of the fuzzy control apparatus is:

IF x_1 is A_1^j and x_2 is A_2^j THEN c is B^m

In which, $X = [x_1, x_2]$ is the input variable of the control processor, C is the output variable of the control processor, A_1^j and A_2^j are the fuzzy values defined on X, the input variable:

$$A_1^j = [NB, NS, ZE, PS, PB], \mu_{A_1^j}(x_1), j=1, 2, \dots, 5$$

$$A_2^j = [N, J, P], \mu_{A_2^j}(x_2), j=1, 2, 3$$

In the above formula, $\mu_{A_i^j}(x_i)$ adopts the function.

$$\mu_{A_i^j}(x_i) = \exp(-(x_i - a_{ij})^2 / b_{ij}^2)$$

In which, a_{ij} is the central element of fuzzy quantum membership function, b_{ij} is the scale factor,

B^m is the fuzzy value defined on C, the output variable,

$$c = \{c_1, \dots, c_5\},$$

B^m can be described by the membership function, $\mu_{B^m}(c_i)$:

$$\mu_{B^m}(c_i) = 1$$

If the fuzzy function of K rules defined is:

$$\mu(X) = (A_1^j, A_2^m) / \left(\prod_{i=1}^{15} (\mu_{A_1^j}(x_i), \mu_{A_2^m}(x_2)) \right) \\ j=1, 2, \dots, 5; m=1, 2, 3$$

Then the output quantum of fuzzy control apparatus is:

$$c(X) = \sum_{j=1}^m \mu(X) c_j$$

In which, "m" is the "control rule number".

To introduce the genetic algorithm:

The characteristic parameter of the fuzzy control apparatus $\{a_{ij}, b_{ij}, c_i\}$ adopts optimal code method of multi-parameter linear reflection. At the same time, let us suppose the sub-length of every parameter is: $l_i=8$; and define the central value of membership function of various fuzzy subsets of x_1 and x_2 to differ from each other; scale factors are regulated between $[-1/2, 0.1]$; the central values of the fuzzy subset membership functions of ZE and Z, which belong to x_1 and x_2 , respectively, is regulated between $[-0.5, 0.5]$ (in which, 0

0.5), the areas of seeking optimization of all the parameters are:

$$a_i \in \{a_{i\min}, a_{i\max}\}$$

$$b_i \in \{b_{i\min}, b_{i\max}\}$$

$$c_i \in \{c_{i\min}, c_{i\max}\}$$

Given: $d = \{d_{\min}, d_{\max}\}$, d_{\min} , d_{\max} are all decimal numbers and they correspond to the binary codes (0 and 1), respectively. They have the following relations.

$$d = d_{\min} + \left(\sum_{i=1}^8 i 2^i (d_{\max} - d_{\min}) \right) / 2^8 - 1$$

Then, we have: the code parameters of x_1 are: a_{1j}, b_{1j} ; the code parameters of x_2 are: a_{2j}, b_{2j} ; the code parameters of c are: c_1, \dots, c_5

Thus, there are 15 initial individuals of parameters when connecting all the parameters according to certain rules. And we can take 36 individuals in order to form the initial community. After that, the fitness function is constructed according to the requirements of the capacity index of the controlled targets. And we need to make use of the genetic operation for membership function to optimize. After the heredity from one generation to another, we will obtain the optimized membership function parameters, and the corresponding fuzzy inference rule is optimized as well.

5. REFERENCES

- [1] Fang Shucheng, Wang Dingwei, Fuzzy Mathematical Theory And Optimization Algorithm, Science Press, Peking, 1997
- [2] David E.Culler, Jaswinder Pal Singh, Anoop Gupta. *Parallel Computing and System Structure*. China Machine Press, Peking, 2002
- [3] Albrecht R, Reeves N and Steels N, Artificial neural nets and genetic algorithms, New York: Springer Verlag, 1993.
- [4] Davis L., Handbook of genetic algorithms, New York: Nostrand Reinhold, 1987.
- [5] Gen M and Liu B. A genetic algorithm for nonlinear goal programming, Technical report, In: ISE95-5, Ashikaga Institute of Technology, Japan, 1995.
- [6] Zhang Ying, Liu Yangiu, software computing, China Science Press, Beijing, 2002
- [7] Zhu Jing, Fuzzy Control Principle and Application, China Machine Press, Beijing, 1995



Zhang Jianhua is a professor and a head of Interface and Control Lab., School of Information & Science and Engineering, Shenyang University of Technology, P. R. China.



Jiang Qian is studying in school of Electrical Engineering, Shenyang University of Technology, P. R. China.

Parallel Multi-grid Algorithm Based on Cluster Computing with Application to Transient Heat Transfer

Zongbo Zhu¹, Guoxun Yang², Chunxiao Liu³, Jinsheng Xiao⁴

¹ Center of Advanced Educational Technology, Wuhan University of Technology, Hubei 430063, China

² School of Computer Science and Technology, Wuhan University of Technology, Hubei 430063, China

³ School of Energy and Power Engineering, Wuhan University of Technology, Hubei 430063, China

⁴ State Key Laboratory of Advanced Technology for Materials Synthesis and Processing, Wuhan University of Technology, Hubei 430070, China

Email: zbzhu@mail.whut.edu.cn Tel: (027)86550621, 13971349939

ABSTRACT

The partial differential equations are often used in science and engineering applications. The solutions cannot be got analytical, so the numerical methods are often used to get approximate results. To achieve high precision, more computing time is needed. But the long time in practical applications is not often allowed, so the precision of final results has to be reduced. Based on cluster system, this paper studies the parallel multi-grid method and its application in the numerical analysis of heat transfer. The results from the sample show that the method not only can expand the size of solved problems efficiently, but also can gain excellent parallel efficiency; therefore it is a method suitable for network parallel environment based on cluster system.

Keywords: parallel algorithm; multi-grid algorithm; heat transfer; heat conduction; domain decomposition methods; cluster computing

1. INTRODUCTION

With the development of scientific research and engineering technology, scientific computing for large-scale physical background is more prevalent. The constraints in the saving and computing competence of the serial system make people turn their eyes on the parallel system. Since the 1990s, the emergence of network cluster system is widely supported and welcomed in the scientific and engineering computing area^[1].

The cluster computing realize a new supercomputing based on network workstation, changed the concept of traditional supercomputing. It considers the cluster system as a modern parallel computer, i.e. SPMD (Single Program Multiple Data), which is different from SIMD and MIMD.

The study on the algorithm and its application in the traditional supercomputer has been executed for several decades, but the development of the algorithm based on cluster computing and its application is just on the beginning. It is a very important problem for cluster computing how to design efficient parallel algorithms and develop parallel application samples that can solve practical problems according to the feature of cluster computing. This paper studies the parallel multi-grid algorithm based on domain decomposition and its application in the numerical analysis of heat transfer under the network parallel cluster environment. The results from the sample show that the method not only can enlarge the size of solved problems efficiently, but also can gain excellent parallel efficiency; therefore it is a method suitable for network parallel environment based on cluster

system.

2. MATHEMATICAL AND NUMERICAL MODEL OF TRANSIENT HEAT TRANSFER PROBLEM

Heat transfer problems includes the contents in heat conduction, heat convection and heat radiation, etc. As for heat conduction problem, the three dimensional heat conduction differential equation of uniform and isotropic medium in cartesian coordinate system can be deduced from Fourier heat conduction law and conservation of energy as^[2]

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) + q_v = \frac{\partial}{\partial t} (\rho c T) \quad (1)$$

where k is the coefficient of heat conduction, $W/(m^{\circ}C)$; ρ is the density, kg/m^3 ; c is specific heat, $J/(kg^{\circ}C)$; q_v is the strength of heat source, i.e. heat quantity produced in unit time and unit volume, W/m^3 ; t is time, s.

In order to conform to the description of the algorithm in mathematics, u represents temperature in place of T in equation (1). Therefore, the differential equation of one-dimensional transient heat conduction problem with constant physical properties and no internal heat source is^[3]

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{a} \frac{\partial u}{\partial t} \quad (2)$$

where $a = k/(\rho c)$ is the thermal diffusivity, m^2/s . The corresponding difference scheme is

$$-Fu_{i-1} + (1 + 2F)u_i - Fu_{i+1} = u_i^0 \quad (3)$$

where $F = a\Delta t/(\Delta x)^2$, is Fourier number with Δt as characteristic time and Δx as characteristic length; u_i^0 is the value of u_i on the previous time.

3. MULTI-GRID METHOD FOR SOLVING TRANSIENT HEAT TRANSFER PROBLEM

The MGM (Multi-Grid Method) is a efficient serial algorithm, its power is derived from the different degree of attenuation of Fourier component error during the iterative solving of discrete equation in elliptic type. The three pillars of MGM are the relaxation of fine grid, the adjustment of coarse grid and the telescopic technology. The basic idea is that to remove the low frequency component (long wavelength component, i.e. smooth error) in terms of the residual error adjustment feature of coarse grids, to remove the high frequency component

(short wavelength component, i.e. vibratory error) in terms of the relaxation smooth feature of fine grids and the telescopic technology is responsible for connecting all grid layers to solve the same problem by restriction and interpolation operator^[4].

The MGM usually includes three basic forms: (1) two-layer V

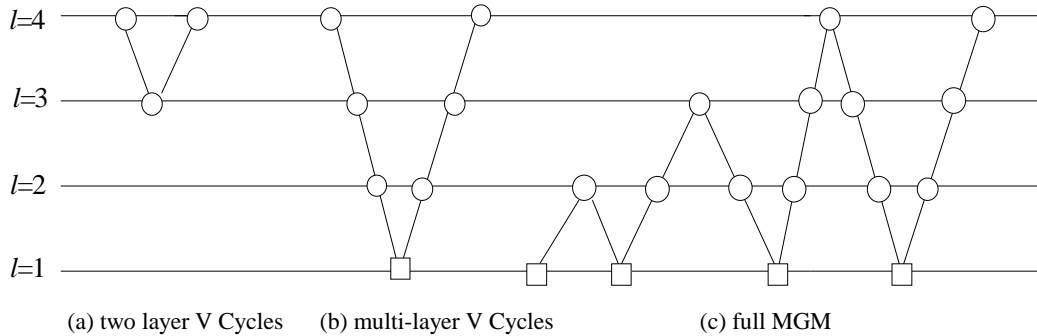


Fig.1 Three basic forms of MGM

Figure 1(a) shows that two-layer V cycle MGM can be attributed as relaxation and iterating the original solution in fine grids to gain residual error, restricting the residual error to coarse grids to gain adjustment value, and interpolating the adjustment value in the fine grids to get adjustment solution. An approximate solution can be obtained through this two-layer V cycle. If the value obtained cannot meet the demand, it is considered as the initial value of u_n and to repeat the above process until the new solution reaches its precision.

From Figure 1(b), we can see that multi-layer V cycle MGM is to restrict the residual error in fine grids to the bottom (the coarsest layer) and get the adjustment value in the bottom accurately, to interpolate the adjustment value on the top (the finest layer) and get the new approximate solution. Multi-layer V cycle MGM can be simplified as “getting the adjustment value while restricting the residual error to the bottom and interpolating the adjustment value on the top to get the solution”.

Figure 1(c) shows that full MGM is to gain the initial solution from the coarsest layer, then gain the solution in the fine layer by interpolate the value upward, and continue interpolating the new value upward to get the solution in the finer layer through the improvement of one or more V cycle. In this way, the solution in the finest layer can be obtained through cycle and rising in wave style. This is a from-coarse-to-fine, interpolating and improving process. As for steady state problem, a satisfied solution can be obtained once through the above process; but as for transient problem, the solution obtained once is of some time. An accurate solution can usually be achieved through one or two V cycle. Figure 1(c) is designed to illustrate full MGM with one V cycle.

4. PARALLEL MULTI-GRID ALGORITHM BASED ON DOMAIN DECOMPOSITION METHOD

cycle MGM; (2) multi-layer V cycle MGM; (3) full MGM. Figure 1 illustrates the above three methods by the example of four layer grids, in which \square represents relaxation and iteration, \backslash represents restrict; $/$ represents interpolation and \circ represents accurate solution.

4.1 Domain decomposition method

The Domain Decomposition Method (DDM) is a new method originated from the 1980s, which integrates the parallel algorithm, the preprocessing technology, the multi-grid multi-level technology and the high-speed algorithm. It can decompose large-scale problem to small-scale problems, complex boundary problem to simple boundary problems, and serial problem to parallel problems.

The DDM is a method that usually to decompose a complex region to lots of subregions according to certain rule as physical properties, geometric shape, discrete modes, feature of the algorithm and the number of processors. Thus, solving the original problem is changed to solving the problem in the subregions^[5,6].

The DDM includes two methods, i.e. decomposed subregions with overlap and without overlap. DDM with overlap is that there is an overlap region for two adjacent subregions. DDM without overlap is that the intersection of two adjacent subregions is null.

The parallel algorithm programming involved in this paper is designed according to master slaver mode based on DDM.

4.2 Parallel multi-grid method

The parallel multi-grid method is introduced in this part with the example of three-layer grids for one-dimensional problem shown in Figure 2. The temperature of nodes on the boundary is known, i.e. the first boundary condition is assumed, for the sake of simpleness and convenience. The parallel computing model with master slaver mode is adopted. The region is decomposed into two subregions, which has a overlap region with one step length. Master creates two Slaves and sends the original data such as the initial condition, boundary condition, etc to each Slave. Each Slave takes charge of solving for each subregion and returns the result to Master to form the solution of the original problem.

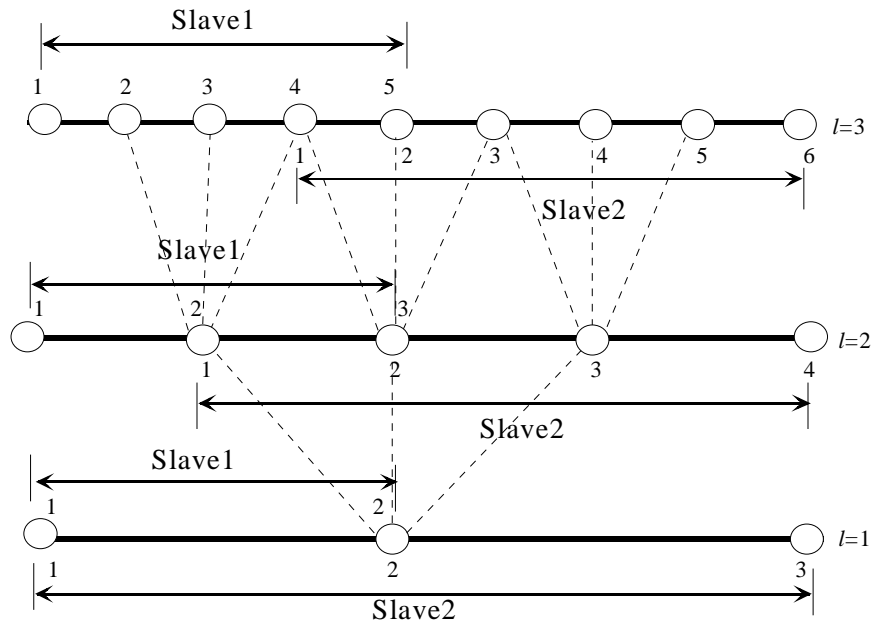


Fig.2 Illustration of data exchange in the parallel algorithm

In the parallel MGM based on DDM, the methods adopted in each subregion are similar to those in serial MGM, i.e. relaxation and iteration is Gauss-seidel iteration with red-black order, the computing method for the residual error is similar to serial MGM. But, interpolation and restrict methods for different subregions varies. Figure 2 shows that the restrict method of Slave1 is the same as that of serial MGM when restricting the residual error of $l=3$ layer to $l=2$ layer, but not for Slave2. Similarly, the interpolation method of Slave1 is the same as that of serial MGM when interpolating $l=3$ layer to $l=2$ layer, but not for Slave2. Slaves with adjacent subregions need data exchange during solving process. MGM includes four processes, i.e. relaxation and iteration, computing the residual error, restrict and interpolation. Therefore, the key point of parallel MGM lies in the data exchange among the four processes.

The data exchange among the four processes for two Slaves is shown in Figure 2.

(1) Relaxation and iteration: The data exchange in relaxation and iteration is introduced for three-layer grids in Fig.2. The value of nodes on the boundary can usually be obtained from the boundary conditions in the serial algorithm, therefore iteration only proceeds among the internal nodes. But for the parallel algorithm based on DDM, node 5 in Slave1 is a virtual node on the right boundary; node 1 in Slave2 is a virtual node on the left boundary. Therefore, when the relaxation and iteration happens between Slave1 and Slave2, it does in fact happen among node 2, 3, 4 in Slave1 and node 2, 3, 4, 5 in Slave2. That is to say, the value of node 1, 5 in Slave1 and node 1, 6 in Slave2 doesn't need to be computed. Obviously, data exchange is needed between Slave1 and Slave2 after one iteration, i.e. the value of node 4 is sent by Slave1 to Slave2 as the value of node 1, which is the virtual node on the left boundary, and the value of node 2 is sent by Slave2 to Slave1 as the value of node 5 that is the virtual node on the right boundary.

As has been said before, the Gauss-Seidel iteration method

with red-black order is still adopted in the relaxation and iteration process. Figure 2 shows that the value of virtual node on the right boundary can be obtained after the iteration of red point (even node). If there are two Slaves, the value of virtual node on the left boundary can also be obtained; and if there are more Slaves, it can be obtained after the iteration of all black points (odd nodes).

(2) Residual error: The temperature of node with the first boundary conditions is known in serial algorithm, so its residual error is zero. As a result, the residual error of virtual node on the boundary is assumed as zero in parallel algorithm. But, its actual residual error is not zero. Thereby, the data exchange is also needed in the residual error computing process. For example, while computing the residual error of the third layer in Figure 2, the residual error of node 4 in Slave1 is sent to Slave2 as the value of node 1, and the value of node 2 in Slave2 is sent to Slave1 as the value of node 5. The data exchange in the residual error computing process is similar to that in relaxation and iteration.

(3) Restrict: Restricting the residual error of $l=3$ layer (fine grid layer) to $l=2$ layer (coarse grid layer) for Slave1 in Fig.2 means restricting the value of node 2, 3, 4 in fine grid layer to node 2 in coarse grid layer and the value of boundary node 1 in fine grid layer to boundary node 1 in coarse grid layer. As for Slave2, it is to restrict the value of node 1, 2, 3 and node 3, 4, 5 in the fine grid layer respectively to node 2, 3 in coarse grid layer, and the value of boundary node 6 to the boundary node 4 in coarse grid layer. At the same time, the data exchange in coarse grid layer is needed, i.e. the residual error of node 2 in Slave1 is sent to Slave2 as the value of node 1, and the value of node 2 in Slave2 is sent to Slave1 as the value of node 3.

(4) Interpolation: In order to interpolate the adjustment value of $l=2$ layer into $l=3$ layer shown in Figure 2, it is for Slave1 that to sent the adjustment value of boundary node 1, 3 in coarse grid layer respectively to boundary node 1, 5 in fine grid layer and the value of node 2 in coarse grid layer to node

3 in fine grid layer, then the value of internal node 2, 4 in fine grid layer can be attained by interpolation. It is for Slave2 that to sent the adjustment value of node 2, 3, 4 in coarse grid layer respectively to node 2, 4, 6 in fine grid layer, then the value of internal node 3, 5 in fine grid layer can be attained by interpolation. At the same time, the data exchange in fine grid layer is also needed, i.e. to send the adjustment value of internal node 4 in Slave1 to Slave2 as the value of virtual node 1 on the left boundary and the value of internal node 2 in Slave2 to Slave1 as the value of virtual node 5 on the right boundary.

In addition, the above process can be modified properly according to different parallel computing environment. If the communication efficiency is low, the execution speed of the process is affected because of the over frequency of data exchange. We can make data exchange happen only in relaxation and iteration process in order to acquire high speed but a result with less accuracy.

5. APPLICATION TO TRANSIENT HEAT TRANSFER PROBLEM

The MGM with two-layer V cycle and multi-layer V cycle, full MGM and parallel MGM are realized for above one-dimensional steady state as well as transient heat conduction problems.

Based on one-dimensional transient heat conduction problem for plane plate with uniform material, the above algorithms are analyzed. The differential equation is equation 2; its difference form is equation 3; and non-dimensional format is adopted. The coefficient of thermal diffusion a is assumed as 1, the thickness of the plate as 2, its initial temperature as 1, and the temperature on the two boundaries of the plate are maintained as 0. Considering the transient temperature distribution with time variable changing from 0 to 1, the finite difference method is adopted to decompose the region. The number of nodes on the grids of the finest layer (tenth layer) is 1025, the corresponding space step length is $1/512$; the number of time steps is 1000, the corresponding time step length is 0.001. The computed result of middle point in the space area is compared with the accurate solution. The iteration control precision (the absolute value of difference between the numerical solution and the accurate solution) for different multi-grid cycle within the same time step is 0.01.

Considering a simple transient heat conduction problem to compare the precision among the algorithms. A plate with thickness as 1, physical properties as 1, initial temperature as 1, thermal insulation on the left boundary, temperature on the right boundary as 0, so its analytic solution is ^[7]

$$u(x, t) = \sum_{n=0}^{\infty} \frac{2(-1)^n}{(n + \frac{1}{2})\pi} \exp\left[-\left(n + \frac{1}{2}\right)^2 \pi^2 t\right] \cos\left(n + \frac{1}{2}\right)\pi x \quad (4)$$

The analytic solution of equation 4 can be considered the accurate solution of temperature field at the range of $0 \leq x \leq 1$. Equation 4 shows that the analytic solution possesses the symmetry with regard to y-axis, therefore its application scope can be extended to $-1 \leq x_1 \leq 1$.

Considering $x = x_1 - 1$, the scope of application for the

analytic solution changes to $0 \leq x \leq 2$ as the accurate solution of this numerical example.

The solutions of parallel MGM in transient heat transfer are shown in Table 1. The solutions are listed at an interval of 100 time steps in the tables. The cycles times in the tables is the average times in a time step for the total times of MGM cycle happened in 100 time steps, and the control cycle times is its maximal value. When the times of MGM cycle reaches the control cycle times in a time step, MGM cycle iteration stops even if the precision of the solution doesn't reach the above control precision. The average running time is the running time in a time step for the total running time happened in 100 time steps. The numerical computing is carried out in Pentium II computer with master frequency 233MHz and internal memory storage 64MB.

The solutions of parallel MGMs in heat transfer are shown in Table 1. The parallel algorithms are realized based on DDM under the Linux operating system and PVM parallel computing environment. Table 2 shows that the solutions of parallel MGM possesses the same characteristics as those of serial MGMs. For computing precision and iteration times, complete MGM is superior to MGM with multi-layer V Cycles. The testing of the performances of the parallel algorithms such as the running time and the acceleration ratio, awaits for network parallel computing environment with high communication efficiency.

6. CONCLUSIONS

Numerical computing experiment proceeds at the application in one-dimensional steady state and transient heat conduction problems by serial and parallel MGM. We can arrive at the following conclusions:

- (1) As for the physical problem with multi-layer grids, full MGM is superior to MGM with multi-layer V cycle, and MGM with multi-layer V cycle is superior to MGM with two-layer V cycle with regards to the iteration times and the precision of the results for the same problem. As a result, full MGM is a very efficient MGM.
- (2) The parallel MGM based on DDM can solve the problem, which cannot be solved by serial algorithm in individual computer.
- (3) The solutions from the parallel MGM possesses the same characteristics as the serial algorithm, i.e. full MGM is superior to MGM with multi-layer V cycle with regards to the precision of the solutions and the iteration times. If the communication efficiency of the parallel environment is not high, the times of communication should be reduced.
- (4) The MGM is applied in transient heat conduction problem in this paper, and the data exchange among adjacent subregions in the grids is solved by improving the interpolation, restriction and relaxation. Therefore, the parallel MGM is successfully applied in the transient heat conduction problem, and it can be further applied in the numerical analysis of the transient heat conduction in the ceramic coatings.

7. REFERENCES

- [1] Jiachang Sun. Network parallel computation and distributed programming environment [M]. Beijing: Science Press, 1996
- [2] Changming Yu, Heat transfer and its numerical analysis [M]. Beijing: Tsinghua University Press, 1981,1~190
- [3] Jinsheng Xiao, Yaonan Qian, Ruison Lu, Numerical Analysis of Heat Transfer and Thermal Loading of I.C. Engines [A], Proceedings of the Second Asian-Pacific Conference on Computational Mechanics [C], Sydney, N.S.W., Australia,1993,Vol 1,pp641-646
- [4] W. Hackbusch, U. Trottenberg (eds.), Multigrid methods [M], Lecture Notes in Math. 960, Spring-Verlag, 1982
- [5] Qingping Guo, Dennis Parkinson, Jinsheng Xiao, Yakup Paker, Parallel Computing using Domain Decomposition for Cyclical Temperatures in Ceramic/Metal Composites, Eleventh International Conference on Domain Decomposition Methods, The University of Greenwich, July 1998.
- [6] Qingping Guo, Yakup Paker, Dennis Parkinson, Jinsheng Xiao, Performance Evaluation of PVM on PC-LAN Distributed Computing, PVM/MPI '98 International Conference, Liverpool, UK, September 1999
- [7] Jinsheng Xiao, Lucheng Deng, Changming Yu, The variational principle and generalized variational principle for the heat conduction problem [J], Mathematica Applicata, Vol 1(4), pp99-105, 1988

ACKNOWLEDGMENTS

This research is partially funded by the Key Project of Science Foundation from Wuhan University of Technology (No.XJJ2002012) and was supported partially by the National Natural Science Foundation of China (No.59405009) and the Royal Society of UK (No.Q724). The authors would like to give their thanks to Professor Yakup Park, Dr. Dennis Parkinson and Professor Qingping Guo for their support and cooperation in the joint project.

Tab.1 Temperature solutions of transient heat transfer problem by parallel MGMs

Time points	Accurate solutions	Multi-layer V cycle		Full MGM	
		Numerical	Cycle times	Numerical	Cycle times
		solutions		solutions	
0.1	0.949305	0.940437	1.00	0.948600	1.00
0.2	0.772312	0.761731	3.76	0.772583	1.00
0.3	0.606804	0.596787	3.09	0.607375	1.00
0.4	0.474487	0.464490	2.61	0.475119	1.00
0.5	0.370777	0.360780	2.28	0.371400	1.00
0.6	0.289709	0.279722	1.83	0.290293	1.00
0.7	0.226363	0.216373	1.36	0.226895	1.00
0.8	0.176867	0.166951	1.02	0.177343	1.00
0.9	0.138194	0.128772	1.00	0.138612	1.00
1.0	0.107977	0.099323	1.00	0.108340	1.00
Control cycle times		4		1	
Actual precision achieved		0.01		0.001	

An Error Bound for the SAOR Method

¹Liu Futi, ¹Huang Tingzhu, ²He huiming

¹School of Applied Mathematics, University of Electronic Science and Technology of China,
Chengdu, Sichuan, 610054, P. R. China,

E-mail: tzhuang@uestc.edu.cn Tel: 028-83202637

²Yongtai Education Committee, Zhongjiang, Sichuan, 618116, P.R. China

ABSTRACT

Suppose $Ax = b$ is a system of linear equations where the matrix A is symmetric positive definite and consistently ordered. A bound for the norm of the errors $\varepsilon_k = x - x^k$ of the SAOR method in terms of the norms of $\delta_k = x^k - x^{k-1}$ and $\delta_{k+1} = x^{k+1} - x^k$ and their inner product is derived.

Keywords: linear systems; SAOR method; error bound.

1. INTRODUCTION

In order to solve linear systems

$$Ax = b \quad (1.1)$$

where A is an $n \times n$ real nonsingular matrix, the symmetric accelerated overrelaxation (SAOR) method was proposed. If the diagonal elements of the matrix A are nonzero, let the matrix A has the splitting $A = I - L - U$, where L and U are strictly lower and upper triangular matrices of A , respectively. The iterative scheme of the SAOR method is defined by

$$x^{k+1} = S_{\gamma, \omega} x^k + Qb, \quad k=1, 2, \dots, \quad (1.2)$$

where

$$Q = \omega(I - \gamma U)^{-1}[(2 - \omega)I + (\omega - \gamma)(L + U)](I - \gamma L)^{-1},$$

$$S_{\gamma, \omega} = U_{\gamma, \omega} \cdot L_{\gamma, \omega}$$

$$L_{\gamma, \omega} = (I - \gamma L)^{-1}[(1 - \omega)I + (\omega - \gamma)L + \omega U],$$

$$U_{\gamma, \omega} = (I - \gamma U)^{-1}[(1 - \omega)I + (\omega - \gamma)U + \omega L]$$

and γ, ω are real parameters.

When the parameter γ equals ω , the SAOR method reduces to the SSOR method (see [5]).

Let $B = L + U$, where B is the Jacobi iterative matrix, let x be the solution of (1.1) and let $\varepsilon_k = x - x^k$ to denote the "error" vector, i.e., the difference between the n th iterate and the exact solution. We use δ_k for the difference between the n th and $(n-1)$ st iterates; thus

$$\delta_k = x^k - x^{k-1}.$$

Then

$$\varepsilon_{k+1} = S_{\gamma, \omega} \varepsilon_k, \quad \delta_{k+1} = L_{\omega_1, \omega_2, \gamma} \delta_k, \quad \varepsilon_k = (I - S_{\gamma, \omega})^{-1} S_{\gamma, \omega} \delta_k.$$

Assume that the matrices A and B satisfy the two conditions:

A1. A is symmetric and positive definite;

A2. A is consistently ordered.

In view of A1 and A2, we can assume that the matrix A is

$$\begin{bmatrix} I & -S^T \\ -S & I \end{bmatrix}.$$

The corresponding Jacobi iterative matrix B is

$$\begin{bmatrix} 0 & S^T \\ S & 0 \end{bmatrix}. \quad (1.3)$$

We suppose that matrix A satisfies the condition A1 and A2, and denote eigenvalues of B by μ_i , $i = 1, \dots, n$. If all μ_i are real, set

$$\underline{\mu} = \min_{1 \leq i \leq n} \{ \mu_i \}, \quad \bar{\mu} = \max_{1 \leq i \leq n} \{ \mu_i \}.$$

Obviously, if A is positive definite, then μ_i are real, and $\underline{\mu} < 0 < \bar{\mu}$.

Now we state some results of the SAOR method.

Lemma 1.1 ([1]). If matrix A be consistently-ordered and its diagonal elements are nonzero, let eigenvalues of the Jacobi matrix be all real numbers and $\bar{\mu} < 1$, then the SAOR method converges if the parameters ω, γ satisfy

$0 \leq \gamma \leq 2$, $0 < \omega \leq 1 + \frac{\gamma}{4 - \gamma}$, and γ, ω don't equal 2 at the same time.

Lemma 1.2 ([1]). If matrix A be consistently-ordered and its diagonal elements are nonzero, let eigenvalues of $S_{\gamma, \omega}$ and the corresponding Jacobi iterative matrix B be $\{\lambda\}$ and $\{\mu\}$ respectively, then

$$[\lambda - (1 - \omega)^2]^2 = [\omega\gamma(2\omega + 2\gamma - 4\gamma\omega + \omega\gamma^2) - \omega^2(\omega - \gamma)^2\mu^2 + 2\omega(1 - \omega)^2(\omega - \gamma)]\mu^2$$

2. EIGENVALUES AND EIGENVECTORS OF $S_{\gamma, \omega}$

From this section to the end, we only suppose that the assumptions A1 and A2 are satisfied and the SAOR method is convergent. Further, without loss of generality, we can assume that S in (1.3) is a nonsingular matrix of order $m = n/2$. From [2, 5] it is known that the eigenvalues of B are related by $-1 < -\mu_1 \leq -\mu_2 \leq \dots \leq -\mu_m < 0 < \mu_m \leq \dots \leq \mu_2 \leq \mu_1 < 1$.

Let

$$z_i = \begin{pmatrix} z_i^{(1)} \\ z_i^{(2)} \end{pmatrix}, \quad i = 1, 2, \dots, m,$$

be the eigenvectors of B corresponding to μ_i , where $z_i^{(1)}, z_i^{(2)}$ are vectors of the length m which consist of the first m and the last m components of the vector z_i . Then

$$\bar{z}_i = \begin{pmatrix} z_i^{(1)} \\ -z_i^{(2)} \end{pmatrix}, \quad i = 1, 2, \dots, m$$

are the eigenvectors of B corresponding to $-\mu_i$.

Lemma 2.1 ([2]). Suppose Assumption A1 and A2 are

satisfied by a matrix A and the associated Jacobi matrix B . Let z_i be the unit eigenvector of B corresponding to the eigenvalue $\mu_i > 0$. If $z_i^{(1)}, z_i^{(2)}$ are as defined above, then

$$\langle z_i^{(1)}, z_i^{(1)} \rangle = \langle z_i^{(2)}, z_i^{(2)} \rangle = \frac{1}{2}, \quad i = 1, 2, \dots, m,$$

and

$$\langle z_i^{(1)}, z_j^{(1)} \rangle = \langle z_i^{(2)}, z_j^{(2)} \rangle = 0, \quad i \neq j; \quad i, j = 1, 2, \dots, m.$$

By Lemma 1.2, we have for $i = 1, \dots, m$,

$$\lambda_i = \frac{1}{2} [2(1-\omega)^2 + (2\omega(\gamma-\omega) + \omega^2(2-\gamma)^2)\mu_i^2 + \omega(2-\gamma)\mu_i\sqrt{R_i}]$$

$$\bar{\lambda}_i = \frac{1}{2} [2(1-\omega)^2 + (2\omega(\gamma-\omega) + \omega^2(2-\gamma)^2)\mu_i^2 - \omega(2-\gamma)\mu_i\sqrt{R_i}]$$

where $R_i = \omega\mu_i^2(4\gamma - 4\gamma\omega + \omega\gamma^2) + 4(1-\omega)^2$.

Obviously, if $\omega(4\gamma - 4\gamma\omega + \omega\gamma^2) \geq 0$, then $R_i \geq 0$ and if $\omega(4\gamma - 4\gamma\omega + \omega\gamma^2) \leq 0$, $\mu_i^2 < 1$, then

$$R_i \geq \omega(4\gamma - 4\gamma\omega + \omega\gamma^2) + 4(1-\omega)^2 = [\omega\gamma + 2(1-\omega)]^2 \geq 0.$$

Now we construct the eigenvalues and eigenvectors of $S_{\gamma,\omega}$.

Let

$$U_i = \sqrt{2} \begin{pmatrix} z_i^{(1)} \\ \alpha_i z_i^{(2)} \end{pmatrix}, V_i = \sqrt{2} \begin{pmatrix} z_i^{(1)} \\ -\alpha_i z_i^{(2)} \end{pmatrix}, \text{ if } R_i > 0, \quad (2.1)$$

where $\alpha_i = \frac{\omega\gamma\mu_i + \sqrt{R_i}}{2(1-\omega + \omega\gamma\mu_i^2)}$, $\bar{\alpha}_i = \frac{\omega\gamma\mu_i - \sqrt{R_i}}{2(1-\omega + \omega\gamma\mu_i^2)}$,

$i = 1, \dots, m$,

or

$$U_i = \sqrt{2} \begin{pmatrix} z_i^{(1)} \\ \beta_i z_i^{(2)} \end{pmatrix}, V_i = \sqrt{2} \begin{pmatrix} 0 \\ \Delta_i z_i^{(2)} \end{pmatrix}, \text{ if } R_i = 0, \quad (2.2)$$

where $\beta_i = \frac{\omega\gamma\mu_i}{2(1-\omega + \omega\gamma\mu_i^2)}$, $\Delta_i = \frac{1}{2\beta_i}$, $i = 1, \dots, m$.

When B satisfies Assumption A2, $S_{\gamma,\omega}$ is given by

$$S_{\gamma,\omega} = \begin{pmatrix} aI + bS^T S & cS^T + dS^T S S^T \\ eS & aI + fS S^T \end{pmatrix},$$

where $a = (1-\omega)^2$, $b = \omega(\gamma + \omega - 3\omega\gamma + \omega\gamma^2)$, $c = \omega(1-\omega)(2-\gamma)$, $d = \omega^2\gamma(2-\gamma)$, $e = \omega(1-\omega)(2-\gamma)$, $f = \omega(\omega + \gamma - \omega\gamma)$.

By direct calculation, it is easy to prove the following statements.

Lemma 2.2. For $j = 1, \dots, m$, there holds

$$S_{\gamma,\omega} U_j = \lambda_j U_j, \quad S_{\gamma,\omega} V_j = \bar{\lambda}_j V_j, \quad \text{if } R_j > 0,$$

or

$$S_{\gamma,\omega} U_j = \lambda_j U_j, \quad S_{\gamma,\omega} V_j = \lambda_j V_j + n_j U_j, \quad \text{if } R_j = 0,$$

where $n_j = \frac{1}{\gamma}(1-\omega + \omega\gamma\mu_j^2)^2(2-\gamma)$.

Lemma 2.3. Let the definitions of U_j and V_j ($j = 1, \dots, m$) are the same as those in (2.1) and (2.2). Then the set of vectors

$\{U_j, V_j\}$ ($j = 1, \dots, m$) is a basis for C^n . Furthermore,

$$\langle U_i, U_j \rangle = \langle U_i, V_j \rangle = \langle V_i, U_j \rangle = \langle V_i, V_j \rangle = 0, \text{ if } i \neq j;$$

if $R_j > 0$, then

$$\langle U_j, U_j \rangle = 1 + \alpha_j^2, \quad \langle V_j, V_j \rangle = 1 + \bar{\alpha}_j^2,$$

$$\langle U_j, V_j \rangle = \langle V_j, U_j \rangle = \frac{\omega\gamma\mu_j^2}{1-\omega + \omega\gamma\mu_j^2};$$

if $R_j = 0$, then

$$\langle U_j, U_j \rangle = 1 + \beta_j^2, \quad \langle V_j, V_j \rangle = \Delta_j^2,$$

$$\langle U_j, V_j \rangle = \langle V_j, U_j \rangle = \frac{1}{2}.$$

Lemma 2.4. Let U_j and V_j ($j = 1, \dots, m$) be as defined above in (2.1) and (2.2). If a_j, b_j are real numbers, then

$$\langle a_i U_i + b_i V_i, c_j U_j + d_j V_j \rangle = 0, \quad \text{if } i \neq j;$$

also, if $R_j > 0$,

$$\langle a_j U_j + b_j V_j, c_j U_j + d_j V_j \rangle = a_j c_j (1 + \alpha_j^2) + b_j d_j (1 + \bar{\alpha}_j^2) + (a_j d_j + b_j c_j) \frac{\omega\gamma\mu_j^2}{1-\omega + \omega\gamma\mu_j^2};$$

and if $R_j = 0$,

$$\langle a_j U_j + b_j V_j, c_j U_j + d_j V_j \rangle = a_j c_j (1 + \beta_j^2) + b_j d_j \Delta_j^2 + \frac{1}{2} (a_j d_j + b_j c_j).$$

Now we expand ε_k , δ_k , etc., in terms of the basis $\{U_j, V_j\}$, $j = 1, \dots, m$. That is, for some real numbers a_j and b_j , $j = 1, \dots, m$,

$$\delta_k = \sum_{j=1}^m (a_j U_j + b_j V_j) = \sum_{j=1}^m \xi_j,$$

Thus

$$\delta_{k+1} = \sum_{j=1}^m S_{\gamma,\omega} \xi_j, \quad \varepsilon_k = \sum_{j=1}^m (I - S_{\gamma,\omega})^{-1} S_{\gamma,\omega} \xi_j.$$

Since $\langle \xi_i, \xi_j \rangle = 0$ for $i \neq j$, it follows that

$$\|\delta_k\|_2^2 = \sum_{j=1}^m A_j, \quad \langle \delta_k, \delta_{k+1} \rangle = \sum_{j=1}^m B_j,$$

$$\|\delta_{k+1}\|_2^2 = \sum_{j=1}^m C_j, \quad \|\varepsilon_k\|_2^2 = \sum_{j=1}^m E_j,$$

where $A_j = \|\xi_j\|_2^2$, $B_j = \langle \xi_j, S_{\gamma,\omega} \xi_j \rangle$, $C_j = \|S_{\gamma,\omega} \xi_j\|_2^2$,

$$E_j = \|(I - S_{\gamma,\omega})^{-1} S_{\gamma,\omega} \xi_j\|_2^2.$$

3. AN ERROR BOUND

Lemma 3.1. Let A_j , B_j , C_j , and E_j be as above and let

$$D_j = (1 - \lambda_j)(1 - \bar{\lambda}_j), \text{ then}$$

$$E_j D_j^2 = \lambda_j^2 \bar{\lambda}_j^2 A_j - 2 \lambda_j \bar{\lambda}_j B_j + C_j.$$

Proof. Case 1. For $R_j > 0$,

$$A_j = \langle a_j U_j + b_j V_j, a_j U_j + b_j V_j \rangle$$

$$= a_j^2 (1 + \alpha_j^2) + b_j^2 (1 + \bar{\alpha}_j^2) + 2a_j b_j \frac{\omega\gamma\mu_j^2}{1-\omega + \omega\gamma\mu_j^2},$$

$$B_j = \langle a_j U_j + b_j V_j, a_j \lambda_j U_j + b_j \bar{\lambda}_j V_j \rangle$$

$$\begin{aligned}
&= a_j^2 \lambda_j (1 + \alpha_j^2) + b_j^2 \bar{\lambda}_j (1 + \bar{\alpha}_j^2) \\
&\quad + a_j b_j (\lambda_j + \bar{\lambda}_j) \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2}, \\
C_j &= \langle a_j \lambda_j U_j + b_j \bar{\lambda}_j V_j, a_j \lambda_j U_j + b_j \bar{\lambda}_j V_j \rangle \\
&= a_j^2 \lambda_j^2 (1 + \alpha_j^2) + b_j^2 \bar{\lambda}_j^2 (1 + \bar{\alpha}_j^2) \\
&\quad + 2 a_j b_j \lambda_j \bar{\lambda}_j \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2}, \\
E_j &= \left\| \frac{a_j \lambda_j U_j}{1 - \lambda_j} + \frac{b_j \bar{\lambda}_j V_j}{1 - \bar{\lambda}_j} \right\|_2^2 \\
&= \frac{1}{(1 - \lambda_j)^2 (1 - \bar{\lambda}_j)^2} [a_j^2 \lambda_j^2 (1 + \alpha_j^2) (1 - \bar{\lambda}_j)^2 \\
&\quad + b_j^2 \bar{\lambda}_j^2 (1 + \bar{\alpha}_j^2) (1 - \lambda_j)^2 \\
&\quad + 2 a_j b_j \lambda_j \bar{\lambda}_j (1 - \lambda_j) (1 - \bar{\lambda}_j) \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2}].
\end{aligned}$$

It follows that

$$\begin{aligned}
E_j D_j^2 &= a_j^2 \lambda_j^2 (1 + \alpha_j^2) (1 - \bar{\lambda}_j)^2 + b_j^2 \bar{\lambda}_j^2 (1 + \bar{\alpha}_j^2) (1 - \lambda_j)^2 \\
&\quad + 2 a_j b_j \lambda_j \bar{\lambda}_j (1 - \lambda_j) (1 - \bar{\lambda}_j) \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2} \\
&= a_j^2 (1 + \alpha_j^2) [\lambda_j^2 (1 - \bar{\lambda}_j)^2] + b_j^2 (1 + \bar{\alpha}_j^2) [\bar{\lambda}_j^2 (1 - \lambda_j)^2] \\
&\quad + a_j b_j [2 \lambda_j \bar{\lambda}_j (1 - \lambda_j) (1 - \bar{\lambda}_j)] \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2} \\
&= a_j b_j [2 \lambda_j^2 \bar{\lambda}_j^2 - 2 \lambda_j \bar{\lambda}_j (\lambda_j + \bar{\lambda}_j) + 2 \lambda_j \bar{\lambda}_j] \frac{\omega \gamma \mu_j^2}{1 - \omega + \omega \gamma \mu_j^2} \\
&\quad + a_j^2 (1 + \alpha_j^2) (\lambda_j^2 \bar{\lambda}_j^2 - 2 \lambda_j^2 \bar{\lambda}_j + \lambda_j^2) \\
&\quad + b_j^2 (1 + \bar{\alpha}_j^2) (\lambda_j^2 \bar{\lambda}_j^2 - 2 \bar{\lambda}_j^2 \lambda_j + \bar{\lambda}_j^2) \\
&= \lambda_j^2 \bar{\lambda}_j^2 A_j - 2 \lambda_j \bar{\lambda}_j B_j + C_j.
\end{aligned}$$

Case 2: For $R_j = 0$, in this case,

$$\begin{aligned}
A_j &= a_j^2 (1 + \beta_j^2) + b_j^2 \Delta_j^2 + a_j b_j \\
B_j &= \langle a_j U_j + b_j V_j, (a_j \lambda_j + b_j n_j) U_j + b_j \lambda_j V_j \rangle \\
&= (a_j^2 \lambda_j + a_j b_j n_j) (1 + \beta_j^2) + b_j^2 \lambda_j \Delta_j^2 \\
&\quad + \frac{1}{2} (2 a_j b_j \lambda_j + b_j^2 n_j) \\
C_j &= \|(a_j \lambda_j + b_j n_j) U_j + b_j \lambda_j V_j\|_2^2 \\
&= (a_j \lambda_j + b_j n_j)^2 (1 + \beta_j^2) \\
&\quad + b_j^2 \lambda_j^2 \Delta_j^2 + (a_j \lambda_j + b_j n_j) b_j \lambda_j \\
E_j &= \|(a_j \lambda_j + b_j n_j) (1 - \lambda_j) + b_j n_j \lambda_j\| \frac{1}{(1 - \lambda_j)^2} U_j \\
&\quad + \frac{b_j \lambda_j}{1 - \lambda_j} V_j\|_2^2 \\
&= [(a_j \lambda_j + b_j n_j) (1 - \lambda_j) + b_j n_j \lambda_j]^2 \frac{1 + \beta_j^2}{(1 - \lambda_j)^4} \\
&\quad + \frac{b_j^2 \lambda_j^2}{(1 - \lambda_j)^2} \Delta_j^2
\end{aligned}$$

$$+ [(a_j \lambda_j + b_j n_j) (1 - \lambda_j) + b_j n_j \lambda_j] \frac{b_j \lambda_j}{(1 - \lambda_j)^3}$$

Since $R_j = 0$, $\lambda_j = \bar{\lambda}_j$ and $D_j = (1 - \lambda_j)^2$, it follows that

$$\begin{aligned}
E_j D_j^2 &= [(a_j \lambda_j + b_j n_j) (1 - \lambda_j) + b_j n_j \lambda_j]^2 (1 + \beta_j^2) \\
&\quad + b_j^2 \lambda_j^2 (1 - \lambda_j)^2 \Delta_j^2 \\
&\quad + [(a_j \lambda_j + b_j n_j) (1 - \lambda_j) + b_j n_j \lambda_j] b_j \lambda_j (1 - \lambda_j) \\
&= \lambda_j^4 A_j - 2 \lambda_j^2 B_j + C_j.
\end{aligned}$$

By Lemma 3.1, we can get the following error bound.

Theorem 3.2. Let Assumption A1 and A2 be satisfied by a matrix A and its associated Jacobi iterative matrix B . When the parameters ω, γ satisfy $0 \leq \gamma \leq 2$, $0 < \omega \leq 1 + \frac{\gamma}{4 - \gamma}$, and γ, ω don't equal 2 at the same time, the error vector ε_k of the SAOR method satisfies

$$\begin{aligned}
\|\varepsilon_k\|_2^2 &\leq \frac{1}{\alpha^2} \{ [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_1^2]^4 \|\delta_k\|_2^2 \\
&\quad - 2 [(1 - \omega)^4 + \omega^2 (\gamma - \omega)^2 \mu_1^4] \langle \delta_k, \delta_{k+1} \rangle \\
&\quad + 4 (1 - \omega)^2 \omega(\gamma - \omega) \mu_1^2 \|\delta_k\|_2 \|\delta_{k+1}\|_2 \\
&\quad + \|\delta_{k+1}\|_2^2 \}. \quad (3.1)
\end{aligned}$$

where $\alpha = \omega^2[(\omega - \gamma)^2(1 - \mu_1^2)^2 + (2 - \gamma)(\gamma - 2\omega + 2)(1 - \mu_1^2)]$ and $\delta_k, \varepsilon_k, \delta_{k+1}$ are as de-fined above, γ, ω are the relaxation parameters and μ_1 is the largest eigenvalue of B .

Proof. By Lemma 1.2 it is easy to verify

$$\begin{aligned}
\lambda_j \bar{\lambda}_j &= [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_j^2]^2, \text{ then} \\
E_j D_j^2 &= [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_j^2]^4 A_j \\
&\quad - 2 [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_j^2]^2 B_j + C_j \\
D_j &= (1 - \lambda_j)(1 - \bar{\lambda}_j) \\
&= \omega^2 [(\omega - \gamma)^2 (1 - \mu_j^2)^2 + (2 - \gamma)(\gamma - 2\omega + 2)(1 - \mu_j^2)].
\end{aligned}$$

When the parameters ω, γ satisfy $0 \leq \gamma \leq 2$,

$0 < \omega \leq 1 + \frac{\gamma}{4 - \gamma}$, and γ, ω don't equal 2 at the same time,

we can know that $2 - \gamma \geq 0$ and $\gamma - 2\omega + 2 \geq 0$, and note that $D_j > 0$ when the iterative matrix $S_{\gamma, \omega}$ is convergent, thus $D_i \leq D_j$ if and only if $\mu_i \geq \mu_j \geq 0$. thus $0 < \alpha \leq D_i$ for all i and hence,

$$\begin{aligned}
\sum_{j=1}^m E_j &\leq \frac{1}{\alpha^2} \{ \sum_{j=1}^m A_j [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_j^2]^4 \\
&\quad - 2 \sum_{j=1}^m B_j [(1 - \omega)^2 + \omega(\gamma - \omega) \mu_j^2]^2 + \sum_{j=1}^m C_j \} \quad (3.2)
\end{aligned}$$

Notice that $B_j = \langle \xi_j, S_{\gamma, \omega} \xi_j \rangle$,

$$|B_j| = |\langle \xi_j, S_{\gamma, \omega} \xi_j \rangle| \leq \|\xi_j\|_2 \|S_{\gamma, \omega} \xi_j\|_2,$$

and by the Cauchy-Schwarz inequality,

$$\begin{aligned}
\sum_{j=1}^m |B_j| &\leq \sum_{j=1}^m \|\xi_j\|_2 \|S_{\gamma, \omega} \xi_j\|_2 \\
&\leq \left(\sum_{j=1}^m A_j \right)^{\frac{1}{2}} \left(\sum_{j=1}^m C_j \right)^{\frac{1}{2}}
\end{aligned}$$

$$= \|\delta_k\|_2 \|\delta_{k+1}\|_2. \quad (3.3)$$

Now, from (3.2) and (3.3), we derive

$$\begin{aligned} \sum_{j=1}^m E_j &\leq \frac{1}{\alpha^2} \{ [(1-\omega)^2 + \omega(\gamma-\omega)\mu_j^4] \sum_{j=1}^m A_j \\ &\quad - 2[(1-\omega)^2 + \omega(\gamma-\omega)\mu_j^2] \sum_{j=1}^m B_j + \sum_{j=1}^m C_j \} \\ &\leq \frac{1}{\alpha^2} \{ [(1-\omega)^2 + \omega(\gamma-\omega)\mu_1^4] \|\delta_k\|_2^2 \\ &\quad - 2[(1-\omega)^4 + \omega^2(\gamma-\omega)^2\mu_1^4] \langle \delta_k, \delta_{k+1} \rangle \\ &\quad + 4(1-\omega)^2 |\omega(\gamma-\omega)| \mu_1^2 \|\delta_k\|_2 \|\delta_{k+1}\|_2 \\ &\quad + \|\delta_{k+1}\|_2^2 \}. \end{aligned}$$

Remark. From the inequality (3.1):

(1) For $\omega = \gamma$, the SSOR case, the error bound reduces to

$$\begin{aligned} \|\varepsilon_k\|_2^2 &\leq \frac{1}{\alpha^2} [(\omega-1)^8 \|\delta_k\|_2^2 - 2(\omega-1)^4 \langle \delta_k, \delta_{k+1} \rangle \\ &\quad + \|\delta_{k+1}\|_2^2], \end{aligned}$$

(2) If there is a norm $\|\cdot\|$ such that $\|B\| < 1$, then in the error bound μ_1 can be replaced by $\|B\|$.

4. EXAMPLE

For the Laplace equation

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} &= 0, \quad (x, y) \in \Omega; \\ u|_{\partial\Omega} &= f(x, y), \quad (x, y) \in \partial\Omega, \end{aligned}$$

We use the five-point difference scheme, and take the region Ω and the numbering of the mesh points as in Figure 1. The discretized equation is $Ax=b$, where

$$A = \begin{bmatrix} 4 & 0 & 0 & 0 & -1 & -1 \\ 0 & 4 & 0 & 0 & -1 & -1 \\ 0 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 \\ -1 & -1 & -1 & 0 & 4 & 0 \\ -1 & -1 & 0 & 0 & 0 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 4 \\ 4 \\ 4 \\ 4 \\ 4 \\ 4 \end{bmatrix}$$

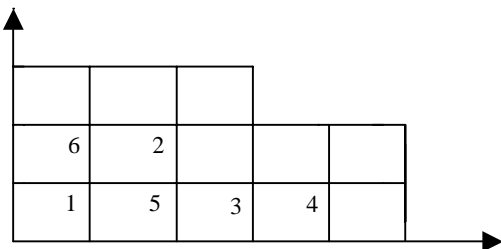


Fig. 1

The matrix A satisfied the assumption A1 and A2. The spectral radius of B is

$$\mu_1 = 5.438319367902683e-001.$$

Using the SAOR method with different parameter pairs $(\omega, \omega_2, \gamma)$ to solve this equation, Table 1 gives the results comparing $\|\varepsilon_k\|$ with the bound given by Theorem 3.2.

Table 1

ω	γ	k	φ_k	$\ \varepsilon_k\ $
----------	----------	-----	-------------	---------------------

0.5	0.1	47	405032005105601e-011	6.563106619260277e-011
0.5	0.6	38	691635218754350e-010	5.293323562020483e-010
0.8	0.4	31	12606116401834e-013	1.631755932194061e-013
1.0	1.0	19	87310895495106e-010	3.687306481070235e-010
1.1	1.0	16	21220143235599e-008	2.707727287523824e-008
1.1	1.2	15	99392168675677e-008	7.826490859219998e-008
1.1	1.09	10	2.899021011413466e-05	2.897571333240592e-005
1.3	0.8	31	10560058992384e-010	5.684871676043075e-010
0.8	1.0	28	20450782870343e-013	4.211682223320247e-013
0.4	0.8	74	16204394631078e-015	1.256073966947020e-015

where

$$\begin{aligned} \varphi_k &= \frac{1}{\alpha} \{ [(1-\omega)^2 + \omega(\gamma-\omega)\mu_1^4] \|\delta_k\|_2^2 \\ &\quad - 2[(1-\omega)^4 + \omega^2(\gamma-\omega)^2\mu_1^4] \langle \delta_k, \delta_{k+1} \rangle \\ &\quad + 4(1-\omega)^2 |\omega(\gamma-\omega)| \mu_1^2 \|\delta_k\|_2 \|\delta_{k+1}\|_2 \\ &\quad + \|\delta_{k+1}\|_2^2 \}^{\frac{1}{2}}. \end{aligned}$$

5. REFERENCES

- [1] Y. Zhang, On the convergence of the symmetric accelerated overrelaxation (SAOR) method, Math. Num. Sinica (in Chinese) 10 (1988)2, 201-204.
- [2] T. R. Hatcher, An error bound for certain successive overrelaxation schemes, SIAM J. Num. Anal., 19(1982), 930-941.
- [3] Y. Z. Song, A note on an error bound for the AOR method, BIT, 39(1999)2, 373-383.
- [4] M. M. Martins, M. Estela, and M. M. Santos, An error bound for the SSOR and USSOR methods, Lin. Alg. Appl., 232 (1996), 131-147.
- [5] D.M.Young, Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.



scientific computing.

Liu Futui is a Lecturer of School of Applied Math of University of Electronic Science and Technology of China. He graduated from School of Applied mathematics with Master and is now pursuing Ph.D. at School of Computer Sci. and Eng., University of Electronic Science and Technology of China. He has published several journal papers. His research interest is



Huang Tingzhu is a Professor, Dr. and the Dean of School of Applied Math of University of Electronic Science and Technology of China. He graduated from Dept of Math, Xi'an Jiaotong University in 2001 with Ph.D.. He was a Visiting Professor of Appl. Math Institute of Chinese Science Academia (1999), a Visiting Professor of Hong Kong Baptist University (2002). He has published several books and over 80 journal papers. His research interests are scientific computing, numerical algebra, and matrix analysis with applications etc.

Analysis of Parallel Matrix Multiplication Algorithms

Wanlong Liu Yaolin Gu
Southern Yangtze University

Wuxi, Jiangsu, 214036, PR of China

Email: lwqjif@hotmail.com , gyl627@sytu.edu.cn Tel. 0510-5868476, 0510-5863635

ABSTRACT

Matrix multiplication is the fundamental operation in many numerical linear algebra applications. Its efficient implementation on parallel high performance computers, together with the implementation of other basic linear algebra operations, is an issue of primary importance for providing these systems with scientific software libraries. Consequently, considerable effort has been devoted to development of efficient practical parallel matrix multiplication algorithms. In this paper, we describe performance analysis of a simple parallel algorithm, Cannon's algorithm, systolic algorithm and hyper-systolic algorithm. Theoretical analysis indicates that the performance of the hyper-systolic algorithm outperforms the other algorithms.

Keywords Matrix Multiplication, Performance Analysis, Cannon's Algorithm, Systolic Algorithm, Hyper-systolic Algorithm

1. INTRODUCTION

This paper discusses parallel algorithms for multiplying two $n \times n$ dense, square matrices A and B to yield the product matrix $C=A \times B$. A concept that is useful in matrix multiplication as well as in a variety of other matrix algorithms is that of block matrix operations [1]. We can often express a matrix computation involving scalar algebraic operations on all its elements in terms of identical matrix algebraic operations on blocks or submatrices of the original matrix. Such algebraic operations on the submatrices are called block matrix operations. For example, a $n \times n$ matrix A can be regarded as an $q \times q$ array of blocks $A_{i,j} (0 \leq i, j < q)$ such that each block is a $(n/q) \times (n/q)$ submatrix. We can use p processes to implement the block version of matrix multiplication in parallel by choosing $q = \sqrt{p}$ and computing a distinct $C_{i,j}$ block at each process.

2. A SIMPLE PARALLEL ALGORITHM

Consider two $n \times n$ matrices A and B partitioned into p blocks $A_{i,j}$ and $B_{i,j} (0 \leq i, j < \sqrt{p})$ of sizes $(n/\sqrt{p}) \times (n/\sqrt{p})$ each. These blocks are mapped onto a $\sqrt{p} \times \sqrt{p}$ logical mesh of processes. The processes are labeled from $P_{0,0}$ to $P_{\sqrt{p}-1, \sqrt{p}-1}$. Process $P_{i,j}$ initially stores $A_{i,j}$ and $B_{i,j}$ and computes block $C_{i,j}$ of the result matrix. Computing submatrix $C_{i,j}$ requires all submatrices

$A_{i,k}$ and $B_{k,j}$ for $0 \leq k < \sqrt{p}$. To acquire all the required blocks, an all-to-all broadcast of matrix A's blocks is performed in each row of processes, and an all-to-all broadcast of matrix B's blocks is performed in each column. After $P_{i,j}$ acquires $A_{i,0}, A_{i,1}, \dots, A_{i, \sqrt{p}-1}$ and $B_{0,j}, B_{1,j}, \dots, B_{\sqrt{p}-1,j}$, it performs the submatrix multiplication and addition step of lines and in Algorithm 1.

Algorithm 1: The block matrix multiplication algorithm for $n \times n$ matrices with a block size of $(n/q) \times (n/q)$

```

procedure BLOCK_MAT_MULT (A, B, C)
begin
  for  $i := 0$  to  $q - 1$  do
    for  $j := 0$  to  $q - 1$  do
      begin
        Initialize all elements of  $C_{i,j}$  to zero;
        for  $k := 0$  to  $q - 1$  do
           $C_{i,j} := C_{i,j} + A_{i,k} \times B_{k,j}$ ;
        end for;
      end BLOCK_MAT_MULT

```

Performance and scalability analysis

The algorithm requires two all-to-all broadcast steps among groups of \sqrt{p} processes. The messages consist of submatrices of n^2/p elements. The total communication time is $2(t_s + \log \sqrt{p} + t_w(n^2/p)(\sqrt{p} - 1))$. After the communication step, each process computes a submatrix $C_{i,j}$, which requires \sqrt{p} multiplications of $(n/\sqrt{p}) \times (n/\sqrt{p})$ submatrices (lines and of Algorithm 1 with $q = \sqrt{p}$). This takes a total of $\sqrt{p} \times (n/\sqrt{p})^3 = n^3/p$ time. Thus, the parallel run time is approximately

$$T_p = n^3/p + t_s \log p + 2t_w n^2 \sqrt{p} \quad (1)$$

The process-time product is $n^3 + t_s p \log p + 2t_w n^2 \sqrt{p}$, and the parallel algorithm is cost-optimal for $p = O(n^2)$. The isoefficiency functions due to t_s and t_w are $t_s p \log p$ and $8t_w^3 p^{3/2}$, respectively. Hence, the overall is efficiency function due to the communication overhead is $O(p^{3/2})$. This algorithm can use a maximum of n^2 processes; hence, $p \leq n^2$ or $n^3 \geq p^{3/2}$. Therefore, the isoefficiency function due to concurrency is also $O(p^{3/2})$.

A notable drawback of this algorithm is its excessive

memory requirements. At the end of the communication phase, each process has \sqrt{p} blocks of both matrices A and B. Since each block requires $O(n^2/p)$ memory, each process requires $O(n^2/\sqrt{p})$ memory. The total memory requirement over all the processes is $O(n^2\sqrt{p})$, which is \sqrt{p} times the memory requirement of the sequential algorithm.

3. CANNON'S ALGORITHM

Cannon's algorithm is a memory-efficient version of the simple algorithm presented in Section 2.

Algorithm2: cannon's algorithm

Shift A_{ij} cyclically left by i steps.

Shift B_{ij} cyclically up by j steps.

locally multiply and accumulate.

Cyclically shift A one step left.

Cyclically shift B one step up.

If not finished goto .

We partition matrices A and B into p square blocks. We label the processes from $P_{0,0}$ to $P_{\sqrt{p}-1,\sqrt{p}-1}$, and initially assign submatrices $A_{i,j}$ and $B_{i,j}$ to process $P_{i,j}$. Although every process in the i^{th} row requires all \sqrt{p} submatrices $A_{i,k}$ ($0 \leq k < \sqrt{p}$), it is possible to schedule the computations of the \sqrt{p} processes of the i^{th} row such that, at any given time, each process is using a different $A_{i,k}$. These blocks can be systematically rotated among the processes after every submatrix multiplication so that every process gets a fresh $A_{i,k}$ after each rotation. If an identical schedule is applied to the columns, then no process holds more than one block of each matrix at any time, and the total memory requirement of the algorithm over all the processes is $O(n^2)$. Cannon's algorithm is based on this idea.

Performance analysis

The initial alignment of the two matrices involves a rowwise and a column wise circular shift. In any of these shifts, the maximum distance over which a block shifts is $\sqrt{p} - 1$. The two shift operations require a total of $2(t_s + t_w n^2/p)$ time. Each of the \sqrt{p} single-step shifts in the compute-and-shift phase of the algorithm takes $t_s + t_w n^2/p$ time. Thus, the total communication time during this phase of the algorithm is $2(t_s + t_w n^2/p)\sqrt{p}$. Each process performs \sqrt{p} multiplications of $(n/\sqrt{p}) \times (n/\sqrt{p})$ submatrices. Assuming that a multiplication and addition pair takes unit time, the total time that each process spends in computation is n^3/p . Thus, the approximate overall parallel run time of this algorithm is

$$T_p = n^3/p + 2t_s\sqrt{p} + 2t_w n^2/\sqrt{p} \quad (2)$$

As in the simple algorithm, the isoefficiency function of

Cannon's algorithm is $O(p^{3/2})$. The advantage of Cannon's algorithm lies in its memory efficiency, as it is possible to arrange the computation such that no cell holds more than one block of each matrix. Disadvantages of Cannon's algorithm are pre-skewing and the fact that if we want to use the optimal data layout for matrix-matrix multiplication, for matrix-vector multiplication, one-to-all broadcast operations are required and the distribution of the result vector differs from the input vector [2].

4. HYPER-SYSTOLIC MATRIX MULTIPLICATION

Next we present the general formulation of the systolic and hyper-systolic matrix multiplication in terms of a pseudo-code. The size of the matrices is $p \times p$ and the 1-Dprocessor array consists of p nodes.

4.1 Systolic algorithm

Systolic arrays are cellular automata models of parallel computing structures in which data processing and transfer are pipelined and the cells carry out functions equal load between consecutive communication events [3]. The systolic version of the matrix multiplication of two matrices A and B is given in Algorithm 3. The matrices are represented in skew order.

Algorithm 3: Systolic matrix-matrix multiplication

DO $j = 1, p$

$C = C + CSHIFT(A, DIM = 1, SHIFT = 1 - j) \times$

$\&SPREAD(B(j,:), DIM = 1)$

$A = CSHIFT(A, DIM = 2, SHIFT = 1)$

END DO

The representation of movements and assignments is simplified by introduction of two functions:

Cshift-row: horizontal circular shift of data b a stride of k on a ring of cells numbered from 1 to n,

$$Cshift-row_n^k(a_{j,i}) := a_{j,(i+k-1+n) \bmod n+1}$$

Cshift-row involves interprocessor communication.

cshift-col: vertical circular shift by a stride of k for the vector of n elements within the systolic cells.

cshift-col amounts to assignments of data element of data elements within the processors.

So, refinement the systolic algorithm:

Algorithm 3: Systolic matrix-matrix multiplication

foreach cell $i = 1 : p$

for $j = 1 : p$

for $l = 1 : p$

$$c_{l,i} = c_{l,i} + cshift - col_p^{1-j}(a_{l,i})b_{j,i}$$

end for

for $k = 1 : p$

$$a_{k,i} = cshift - row_p^1(a_{k,i})$$

end for

end for

end foreach

The algorithm is completely regular. Each cell carries out one operation together with a data assignment, followed by a circular shift of the matrix A. This sequence is repeated in each systolic cycle. The skew order is not destroyed during execution of the algorithm. We note that for each cell, inner cell assignment operation is carried out using equal strides within a given step of the parallel algorithm. We have already noticed above that the complexity of the systolic computation is not competitive with Cannon's algorithm. The hyper-systolic version, however, will belong to the same complexity class as Cannon's algorithm.

4.2 Hyper-systolic algorithm

The concept of hyper-systolic algorithm has been introduced in order to reduce the communication overhead of systolic algorithms [4]. The regular bases are given by

$$A_{k=K-1} := \begin{pmatrix} 0, 1, 1, \dots, 1 \\ \vdots \\ 0, K, K, \dots, K \end{pmatrix}_{K-1} \quad (3)$$

$$B_{k=K-1} := \begin{pmatrix} 0, K, K, \dots, K \\ \vdots \\ 0, K, K, \dots, K \end{pmatrix}_{K-1}$$

$$K \times \tilde{K} = n$$

The completeness of a base pair is defined in terms of the h-range of the base, a notion borrowed from additive number theory [5, 6].

We employ the regular bases constructed for the 2-array hyper-systolic system in a slightly modified version:

$$A_{k=K-1} = (0, K, K, L, K) \quad (4)$$

$$B_{k=K-1} = (0, -1, -1, L, -1)$$

$$C_{k=K-1} = (0, 1, 1, L, 1)$$

Algorithm 4: Hyper-systolic matrix- matrix multiplication

$B(j,:) = \text{CSHIFT}(B(j,:), \text{SHIFT} = \text{MOD}(1-j, K))$

DO $j = 1, \tilde{K}-1$

DO $l = 1, K$

$C(:, l) = C(:, l) + \text{CSHIFT}(A, \text{DIM} = 1, \text{SHIFT} = 1 - (j-1) \times$
 $\&K - l) \times \text{SPREAD}(B((j-1) \times K + l, :), \text{DIM} = 1)$

END DO

$A = \text{CSHIFT}(A, \text{DIM} = 2, \text{SHIFT} = K)$

END DO

DO $l = 1, K$

$C(:, l) = C(:, l) + \text{CSHIFT}(A, \text{DIM} = 1, \text{SHIFT} = 1 - (\tilde{K}-1) \times$
 $\&K - l) \times \text{SPREAD}(B((\tilde{K}-1) \times K + l, :), \text{DIM} = 1)$
 END DO

And, refinement the Hyper-systolic algorithm

Algorithm 4: Hyper-systolic matrix- matrix multiplication

foreach cell $i = 1: p$

for $j = 1, p$

$b_{j,i} = \text{cshift} - \text{row}_p^{(1-j) \bmod K}(b_{j,i})$

end for

for $j = 1: \tilde{K}-1$

for $l = 1: K$

for $n = 1: p$

$c_{n,i}^l = c_{n,i}^l + \text{cshift} - \text{col}_p^{1-(j-1)K-l}(a_{n,i})b_{(j-1)K+l,i}$

end for

end for

for $l = 1: p$

$a_{l,i} = \text{cshift} - \text{row}_p^K(a_{l,i})$

end for

end for

for $l = 1: K$

for $n = 1: p$

$c_{n,i}^l = c_{n,i}^l + \text{cshift} - \text{col}_p^{1-(\tilde{K}-1)K-l}(a_{n,i})b_{(\tilde{K}-1)K+l,i}$

end for

end for

for $j = 1: K-1$

for $l = 1: p$

$c_{l,i}^{K-j} = c_{l,i}^{K-j} + \text{cshift} - \text{row}_p^1(c_{l,i}^{K-j+1})$

end for

end for

end foreach

We see that three shift constants are involve, 1, -1 and K, where the second matrix B is shifted into the negative direction. The hyper-systolic matrix multiplication proceeds in three steps.

- Matrix B is shifted $K-1$ times by strides of 1 along the systolic ring and stored as $B^i (0 \leq i \leq K-1)$. As motivated above, for the case of matrix-matrix multiplications, we can spare communication: it suffices to shift B in \tilde{K} row blocks of K rows each, where, within each block, the first row is shifted by a stride of 0 and the last by a stride of $K-1$.
- \tilde{K} Times, the multiplication of A with K rows of the pre-shifted matrix B is carried out. After each step, A is moved to the left by a hyper-systolic shift of stride K. The result is accumulated within K matrices C^i .
- The K intermediate result matrices, C^i , are shifted back according to base C_k , while summed up to the final matrix C. The algorithm is regular. The skew order is not destroyed during execution, and in any stage, only global addresses are required.

Complexity of hyper-systolic algorithm

The gain factor for the matrix multiplication is improved compared to the 2-array problem as matrix B is only partially shifted. The gain factor R which compares the regular hyper-systolic matrix multiplication to the systolic algorithm is

$$R = \frac{p-1}{K + \tilde{K}-1} \approx \frac{\sqrt{p}}{2} \quad (5)$$

Because one needs 1 shift of the full matrix B, \tilde{K} shifts by K of matrix A and again $K-1$ shifts by 1 of matrix C. Therefore, the total number of shifts required is

$$T = K + \tilde{K}-1 \quad (6)$$

The standard systolic computation requires $p-1$ shifted of the matrix A. For $K = \tilde{K}$ we get $R \approx \frac{\sqrt{p}}{2}$.

Comparison to Cannon's algorithm and systolic algorithm.

In order to compare Cannon's algorithm and the hyper-systolic matrix multiplication, we consider a $\sqrt{p} \times \sqrt{p}$ cell array on which Cannon's algorithm is carried out, and a ring array of p processors on which the hyper-systolic matrix multiplication is implemented. The total number of shift operations of Cannon's algorithm is $2\sqrt{p} - 2 = 2K - 2$, while the number of shift operations for the hyper-systolic algorithm was $2K - 1$. Thus, the complexities in terms of circular shift operations (on matrices of equal size) of both algorithms are equal. Since Cannon's algorithm requires pre-skewing of A and B with non-regular interprocessor communication, the hyper-systolic method bears a clear advantage.

The systolic computation of n^2 -problems on a parallel computer of p processors involves $O(np)$ communication events. The hyper-systolic algorithm can reduce the communication overhead to $O(np^{1/2})$, as has been successfully applied for a prototype n^2 -problem, which involves the computation of all n^2 two-body forces for a system of n gravitatively interacting bodies [7]. This progress makes us confident that hyper-systolic processing can be applied to a variety of numerical problems which lead to n^2 computation events. An important application is found in astrophysics where the investigation of the dynamics and evolution of globular clusters is of prime importance [8]. Further examples of applications are protein folding, polymer dynamics, polyelectrolyte, global and local all-nearest neighbors problems, genome analysis, signal processing etc [9].

5. CONCLUSION

The 1-D hyper-systolic matrix multiplication algorithm is a promising alternative to 2-D matrix multiplication algorithms. Exhibiting equal communication overhead as standard methods like the 2-D cannon algorithm, the hyper-systolic algorithm avoids non-regular communication and indexed local addressing. Hence the hyper-systolic matrix multiplication scheme is applicable on any type of parallel system, even on machines that cannot compute indexed addressing. Additionally the alignment for the optimal hyper-systolic algorithm leads to efficient matrix-vector computations as well.

6. REFERENCES

- [1] J. Choi, J.J. dongarra, D.W. Walker, The design of scalable software libraries for distributed memory concurrent computer, in J.J. dongarra, B.Tourancheau (Eds), Environments and Tools for Parallel Scientific Computing, Elsevier, Amsterdam, 1992.
- [2] V. Kumar, A. Grama, A. Gupta, and G. Karypis: Introduction to parallel computing, Redwood City, Benjamin/Cummings, 1994
- [3] N. Petkov, Systolic Parallel Processing, Amsterdam, North-Holland, 1993
- [4] T. Lippert, A. Seyfried, A. Bode, K. Schilling, Hyper-systolic parallel computing, IEEE Trans. on Parallel and Distributed Systems 9(1998) 1.
- [5] M. Djawadi, G.Hofmeister, The postage stamp problem, mainzer seminarberichte, Additive Zahlentheorie 3 (1993) 187
- [6] R.K. Guy, Unsolved Problems in Number Theory, Springer, Berlin, New York, 1994.
- [7] T. Lippert, U. Glaessner, H. Hoeber, G. Ritzenhöfer, K. Schilling, A. Seyfried, Hyper-systolic processing on APE100/quadratics, I. n^2 -loop computations, Int. J. Mod. Phys. C 7(1996)485.
- [8] G. Meylan, D.C. Heggie, Internal Dynamics of Globular Clusters, Preprint <http://xxx.lanl.gov/ps/astvo-ph/9610076>
- [9] Th. Lippert, Hyper-Systolic Parallel Computing – Theory and Applications, Ph. D thesis, University of Groningen, 1998.



Liu Wanlong received Engineering Bachelor from the Computer Science Department of Harbin Institute of Technology in 1998. He is a postgraduate in school of Information Technology, Southern Yangtze University, P. R. China. His research interests include distributed system and parallel computing.



Gu Yaolin : A Full Professor and Deputy Dean of the School of Information Engineering , Southern Yangtze University, P. R. China. He graduated from the Computer Science Department of Shanghai Jiaotong University in 1982. As a Visiting Scholar of the University of Stony Brook in USA from 1994 to 1995, his main research interests are Computer Graphics and Parallel Computing. Now he is a Senior Member of Computer Society, IEEE.

Multi-stage Influence Diagrams Decision Using Genetic Algorithms*

Zhao Yun^{1,2}, Liu Weiyi¹, Li Jin

Department of Computer Science, Yunnan University¹, Yunnan Nationality University²
Yunnan Kunming, China

Email: zhaoyun7907@hotmail.com Tel: (0871) 13330419050

ABSTRACT

We present an approach to the solution of large multi-stage decision problems under uncertainty represented as influence diagrams. This approach differs from the existing IDs evaluating approaches in the way that it exploits all possible solutions simultaneously using Genetic algorithms. It's an approximate algorithm and is easily paralleled. Particularly, we introduce Game Theory to solve some complex realistic decision problems in multi-agent environment.

Keywords: influence diagrams, genetic algorithm, bayesian network, game theory, Nash equilibrium.

1. INTRODUCTION

Influence diagrams (IDs) provide expressive and intuitive representations for an important class of decision problems [Howard and Matheson 1981[1], Shachter 1986[2], Pearl 1988[3]]. Once an ID is constructed, it can be used to derive a policy that specifies what action the decision maker should take for each decision in order to obtain the maximum expected utility. We refer to the computation of the optimal policy for an ID as evaluating the ID. Single-stage IDs are easily evaluated using the present algorithms. In our work, we mainly focus on computing the global policy for large multi-stage IDs.

Many approaches to evaluate an ID are dynamic programming algorithms (Howard & Matheson 1984[4], Shachter 1986, N.L.Zhang 1998[5]). Most of these approaches involved unfolding it into a decision tree and using the "fold-back" algorithm on that tree. Several other algorithms evaluate an ID directly without transforming it into a decision tree. We call them direct evaluation algorithms (Shachter 1986, Shenoy 1992[6], Ndilikilikesha 1994[7]). Recent work is about converting IDs into BNs and applying some inference algorithm to evaluate them (Cooper 1988[8], Shachter & Peot 1992[9], Y.Xiang & C.Ye 2001[10]), which reduced ID evaluation into BN inference problem. But inference in BNs is difficult and exponential for large problems and is still an active research area. When costs of computation are not taking into account, optimal policies can be determined using these programming. But when the costs are not negligible, these approaches may be infeasible.

To reduce computation complexity, a number of researchers

have described iterative approaches to solving ID.

Greedy-based methods are provided (Hecherman 1989[11], Lehner & Sadigh 1993[12]). They reduce the great evaluation cost, but can't guarantee the global optimal solutions. In this paper, we present an approach based on Genetic algorithms that can be used to computing global optimal policies for large multi-stage ID decision problem effectively.

Evaluating an ID means finding an optimal decision for each of its decision nodes. A global solution to the decision problem consists of a series of decisions that maximizes the expected utility. We present an approach to compute global optimal policies for large multi-stage ID. For a decision path combination (a possible optimal solution to the ID), the approach computes the expected utility of it. Then we use a GA-based approach to exploit the global optimal policies for the ID efficiently, which is an evolving process guaranteeing new generation is in some sense an improvement over the previous one. Furthermore, we discuss the decision-making process facing game, and give a method to compute Nash equilibria under the multi-agent ID decision circumstance.

2. SINGLE AGENT DECISION MAKING USING GENETIC ALGORITHMS

2.1 Influence Diagrams

An influence diagram is a DAG representing a sequential decision problem under uncertainty [Howard & Matheson, 1984]. An ID models the subjective beliefs, preference, and available actions from the perspective of a single decision maker. IDs provide a concise graphical formulation of many decision problems and have been studied as a tool for decision making in intelligent systems.

IDs are extensions of Bayesian networks (BNs), which consist of chance nodes only. Nodes in an ID are of three types. Circle shaped chance nodes represent random variables that the decision maker cannot control. Square shaped decision nodes represent decision, i.e., sets of mutually exclusive actions that the decision maker can take. Diamond shaped utility nodes represent the decision maker's preferences in the form of a value function. There is a conditional probability table associated with every chance node in the form $P(x|pa(x))$ (unconditional, if it has no predecessors). The utility node V has an associated value function, $V: \mathbf{v} \rightarrow R$, which may be represented as a table. The set of \mathbf{v} is the set of all possible combinations of values for utility node V 's information predecessors. We give an example of ID as follows.

2.2 Genetic Algorithms

Genetic algorithm (GA) is a probabilistic search algorithm derived from biological sciences. GA simulates natural

*This work is supported by the National Natural Science Foundation of China (Grant No.60263003), the Yunnan Natural Science Foundation (Grant No.2002F0011M), and the Foundation of the Key Laboratory of Intelligent Information Processing, Institute of Computing technology, Chinese Academy of Sciences (Grant No.IIP2002-2).

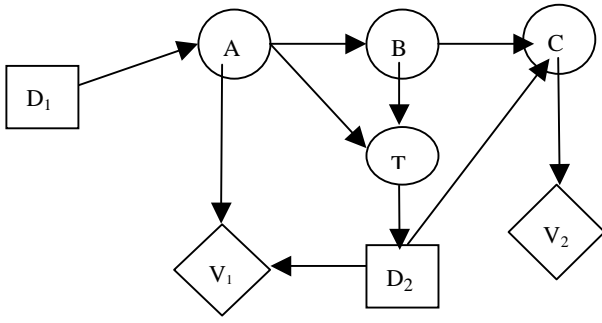


Figure 1. A simple two-stage ID

evolution to find the best solutions by means of natural selection and genetic mechanism. It has become popular tools for search, optimization, machine learning and for solving complex and large-scale problems. Especially, GA is well suited for combination optimization and multipurpose optimization problems, and has been used to solve many NP problems successfully. And GA is easily parallelized. One of the simplest things that can be done is to evaluate the population in parallel. So it's particularly well suited for efficient computation of large multi-stage IDs.

GA starts out with an initial “population” of possible solutions (individuals) to a given problem (environment) where each individual is represented using some form of encoding as a “chromosome”. These chromosomes are evaluated in some way for their “fitness” (i.e. the extent to which the individuals they represent are suitable to the environment). Using their fitness as a criterion, certain chromosomes in the population are selected for reproduction through some stochastic processes such as crossover and mutation, by which new chromosomes can be generated. Each new generation is seen to be in some sense an improvement over the previous one. The process is repeated until we get an acceptable solution to the problem.

The method based on GAs needs not extra domain knowledge, avoiding the limitation of some heuristic search approaches. Furthermore, genetic algorithms climb “multi-hills” in parallel, avoiding getting in local optima. Thirdly, the GA-based methods reduce exponential computation in decision tree evaluation and BN inference for complex problem.

2.3 Computing Global Optimal Policy Using Genetic Algorithm

In an ID, the decision nodes have a temporal order, D_1, \dots, D_n , and the chance nodes are partitioned according to when they are observed: I_0 is the set of chance nodes observed prior to any decision, \dots, I_i is the set of chance nodes observed after D_i is taken and before D_{i+1} is taken. I_n is the set of chance nodes never observed or observed too late to have an impact on any decision. That is, we have a temporal ordering $I_0 < D_1 < I_1 < \dots < D_n < I_n$. In general, previous observations and decisions are relevant and known when making a decision. We need to impose a restriction on the decision problem, namely that a decision cannot have an impact on a variable already observed. This translates into the property $P(I_k | I_0, \dots, I_{k-1}, D_1, \dots, D_n) = P(I_k | I_0, \dots, I_{k-1}, D_1, \dots, D_k)$. In words, we can calculate the joint distribution for I_k without the knowledge of the states of D_{k+1}, \dots, D_n (i.e., the

future decisions).

Each decision node and chance node has finite exclusive states, which we denote as $a_i^{(j)}$ and $s_i^{(j)}$ respectively. $a_i^{(j)}$ refers to the j th action of decision D_i , and $s_i^{(j)}$ refers to the j th state of chance variable I_i . According to the regularity constraint, there exists a directed path that contains all decision nodes. Given a decision (action) path $a_1^{(j)} \dots a_{n-1}^{(k)}$, $a_n^{(i)}$ (i.e., we have already observed the random variables $I_0 \dots I_{n-1}$ and we have chosen alternatives for decisions $D_1 \dots D_n$), we can get the expected utility for action $a_n^{(i)}$ of decision D_n , which is given by

$$EU(a_n^{(i)}) = \sum_{I_n} P(I_n | I_0, K, I_{n-1}, a_1^{(j)}, K, a_{n-1}^{(k)}, a_n^{(i)}) * u(a_n^{(i)} | \gamma) \quad (1)$$

where $u(a_n^{(i)} | \gamma)$ is the utility of taking action $a_n^{(i)}$ under the present content γ .

The expected utility for action $a_k^{(i)}$ of decision D_k ($k < n$) can be obtained in a similar way:

$$EU(a_k^{(i)}) = \sum_{I_k} P(I_k | I_0, K, I_{k-1}, a_1^{(j)}, K, a_k^{(i)}) * u(a_k^{(i)} | \gamma) \quad (2)$$

Thus, the total expected utility of a decision path $a_1^{(j)}, \dots, a_n^{(i)}$ is:

$$EU = \sum_{i=1}^n EU(a_i^{(m)})$$

Now, we must select an adequate representation and an appropriate evaluation function for applying GAs to this problem successfully. We represent each individual as a string of length n , where n is the number of decision nodes. Each bit represents that action in the corresponding decision is chosen. We can adopt real-value coding form. In this way, an individual denotes a decision path and the total expected utility of the decision series can be easily computed.

Evaluating the goodness of an individual should be based on its value of the total expected utility. An action path with higher expected utility should be better. We can define the fitness of an individual X (an action path) in the following formula:

$$F(X) = \sum_{i=1}^n EU(a_i^{(m)}) \quad (3)$$

After all the individuals have been evaluated, we select some superior individuals, and apply them to genetic operators to create offspring for next generation. The process is repeated until $F(X)$ remains unchangeably in continuous generations. For the DI example in figure 1, the set of variables is partitioned into the sets: $I_0 = \{\Phi\}$, $I_1 = \{T\}$, $I_2 = \{A, B, C\}$.

Suppose decision node D_1 has 3 actions, and D_2 has 2 actions. The coding (32) represents a decision path that choosing the third action in the first-stage decision and the second action in

the second-stage decision.

The fitness of this individual is:

$$F(X) = EU(a_1^{(3)}) + EU(a_2^{(2)}).$$

$$EU(a_2^{(2)}) = \sum_{A,B,C} P(A,B,C | I_0, I_1, a_1^{(3)}, a_2^{(2)}) \\ * u(a_2^{(2)} | \gamma),$$

$$EU(a_1^{(3)}) = \sum_T P(T | I_0, a_1^{(3)}) * u(a_1^{(3)} | \gamma).$$

In an ID, the independence restriction can be verified by checking that, there is no directed path from a decision D_k to a decision D_i ($i < k$), and the same to the chance variables. The probability section in EU can be easily calculated.

3. MULTI-AGENT DECISION MAKING

3.1 Computing Global Optimal Policy after Obtaining Extra Information

An ID models a single agent's decision, without regard to the information from other agents. In a single agent's decision making process, we may get some useful information from other agents' which may influence our decision-making process. In this paper, we look a piece of external information I from other agents' as a random variable with a (conditional) probability table, i.e. a normal chance node. So the new utility function can be formulated as follows:

$$EU(a_k^{(i)}) = \sum_{I_k} P(I_k | I_0, K, I_{k-1}, I, a_1^j, K, a_k^i) \\ * u(a_k^i | \gamma)$$

Whenever information is obtained, we can restart the GA-based decision making process from the "information node" and adjust the following decision process. The GA-based evaluation process is in the same method. In this way, we avoid the dilemma in orthodox approaches. The dilemma is that the decision maker may need to take action on the first decision with some urgency, but all the computational effort could go into finding an action for the last decision node.

3.2 Decision Making in Game Environment

So far we have considered various aspects of single-agent decision-making. But in many cases, the decision-making agent will have to take other agent's action into account. Multi-agent decision-making presents some unique challenges and complications. In this paper, we will consider games in agent decision-making process. We introduce game theory to analysis the problem and provide a GA-based method to compute Nash equilibria efficiently.

Suppose three agents A, B and C are strategically relevant, the decision-making agent A needs to take into account the decision at B and C. A will determines what behavior is rational in this interacting environment. In game environment, an action in A with the maximum expected utility is generally inaccessible because we have to consider other agents' actions. We should select the rational action based on Nash equilibria.

A Nash equilibrium is a list of strategies, one for each player, which means no agents can get a better payoff when unilaterally deviate from the equilibrium. But a game in strategic form does not always have Nash equilibrium if each

player deterministically chooses one of his strategies. Players may instead randomly select from among these pure strategies with certain probabilities. Randomizing one's own choice is called a mixed strategy in game theory, i.e., a probability distribution over the player's actions.

Theorem [13]: Any finite normal form game has at least one (mixed strategy) equilibrium.

But it is hard to compute mixed-strategy equilibria. Finding Nash equilibria may be exponentially hard. Can we find a polynomial-time algorithm? Algorithms for computing Nash equilibria are well studied. Classical path following methods, such as the Lemke-Howson algorithm [14] for two person games, and Scarf-type fixed point algorithm [15] for n-persons games provide globally methods for finding a sample equilibrium. For large problems, these methods may be prohibitive. Jie Bao [16] presents an algorithm using hillclimbing, but which not guaranteed to find a global optimal solution. Finding of all equilibria is even more computationally intensive. With current methods, they are only feasible on small problems. Researchers note that n-player games are computationally much harder than 2-player games. If each player has more than two actions, the complexity will be worse.

In this paper, we present a GA-based algorithm for this problem. Our approach may find one or more approximate equilibria. We focus principally on noncooperative game with complete and perfect information. Here we only provide fitness and individual representation for simplicity.

Problems to be solved:

Multi-players (agents): A, B, C...

The strategies (actions) of agent A denote as: a_1, a_2, \dots, a_n , and the same denotation to the other agents.

A strategy combination (a possible Nash equilibrium, i.e., an individual) denotes as:

$$\{p(a_1) \dots p(a_n), p(b_1) \dots p(b_j), p(c_1) \dots p(c_k), \dots\}$$

Here, we assume agent A has n doable actions; agent B has j doable actions; agent C has k doable actions....

$p(a_1) \dots p(a_n)$ is a mixed strategy of agent A, $p(a_i)$ refers to taking action a_i with probability $p(a_i)$, $p(a_i)$

$$\in [0,1], \sum_{i=1}^n p(a_i) = 1, \text{ and the same to others.}$$

An individual is a Nash equilibrium with $S(\cdot) = 0$:

Regret_i(\cdot): agent i's gain by changing strategy combination (\cdot , \cdot , \cdot), which means agent i's strategy changes to \cdot while the other agents' strategies retain unchangeably, \cdot refers to the other strategies except agent i's strategy in an individual \cdot .

Ui(\cdot): agent i's utility (payoff) under the present strategy combination \cdot .

$$\text{Reg}_i(\cdot) = \max_{\theta_i} (Ui(\cdot, \cdot, \cdot) - Ui(\cdot)),$$

\cdot : denotes agent i's gainable maximum payoff plus when unilaterally deviates its strategy.

Total Regret: $S(\cdot) = \text{sum}(\text{Reg}_i(\cdot))$, which denotes the sum of all agents gainable maximum payoff plus when they change their strategies respectively. If each agent has 0 regret, i.e., S

$(\theta) = 0$, the individual θ is a Nash equilibrium.

Then the fitness $F(\theta)$ of an individual θ may denote as follows:

$$F(\theta) = \frac{1}{S(\theta) + 1},$$

$$F(\theta) = \begin{cases} 1, & \text{if } S(\theta) = 0 \\ 0 < F(\theta) < 1, & \text{if } S(\theta) > 0 \end{cases}$$

Here, $F(\theta) \in [0, 1]$. The larger of the value $F(\theta)$, the better the individual θ is.

We use a $\{0, 1\}$ binary string to represent an individual, and the length is based on the precision we desiring and the number of agents. For example, two agents A and B are in a game. Agent A has three doable actions, and agent B has two doable actions. Suppose the solution must be precise to two places of decimals, each action should use 7 bits to represent ($2^6 < 10^2 < 2^7$). For the problem, we may use $7 \times 3 = 21$ bits and $7 \times 2 = 14$ bits binary string to represent the mixed strategy of agent A and B respectively. Each of the string corresponds to a real value in $[0, 1]$. Then the length of an individual is $21 + 14 = 35$ bits. Other coding forms can be obtained easily in similar way.

The algorithm starts with some random individuals, which are selected and reproduced based on some criterion in subsequent generations. The process is repeated until $F(\theta) = 1$ or a certain value given in advance. Thus we can find multiple approximate equilibria.

4. CONCLUSIONS

We have described an algorithm for agent's decision making in large multi-stage decision problem represented as influence diagrams. The process is in a new fashion that looks a possible decision path as an individual and exploits all individuals simultaneously in one generation. The approach guarantees the global optimal policy and reduces the costs in dynamic programming and inference algorithms. Particularly, we present a method to compute NASH equilibria for multi-agent's decision facing game. The approach may find one or more approximate equilibria in accessible cost.

In this paper, we solve the two problems using GAs and we can see that it's really a perfect algorithm for the problems. GAs actually search the space of all possible solutions simultaneously, quickly identifying and exploiting solutions with high performance. The power of a GA derives the ability to exploit a very large number of structural individuals without the computational burden of explicit calculation and storage.

5. REFERENCES

- [1] In R.Howard, and J.Mathson, editors, Readings on the Principles and Applications of Decision Analysis, pages 719-762. Strategic Decisions Group, Menlo Park, California.
- [2] R.D.Shachter (1986), Evaluating influence diagrams, Operations Research, 34(6), pages 871-822.
- [3] J.Pearl (1988), Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers,

- L o s A l t o s , C A .
- [4] R.Howard, and J.Mathson (1984), Readings on the Principles and Applications of Decision Analysis. CA: Strategic Decision Group.
- [5] N.L.Zhang (1998), Probabilistic inference in influence diagrams, In G.F.Cooper and S.Moral, editors, Proc. 14th Conference on Uncertainty in Artificial Intelligence, pages 514-522.
- [6] P.P.Sheony (1992), Valuation-based systems for Bayesian decision analysis, Operations Research, 40(3), pages 463-484.
- [7] P.Ndilikikesha (1994), Potential influence diagrams, International Journal of Approximate Reasoning, 11, pages 251-285.
- [8] G.F.Cooper (1988), A method for using belief networks as influence diagrams, In R.D.Shachter, T.S.Levitt, L.N.Kanal, and J.F.Lemmer, editors, Proc. 4th Workshop on Uncertainty in Artificial Intelligence, pages 55-63.
- [9] R.D.Shachter, and M.Poet (1992), Decision making using probabilistic inference methods, In Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, pages 276-283.
- [10] Y.Xiang and C.Ye (2001), A simple method to evaluate influence diagrams, In Third International Conference on Cognitive Science.
- [11] D.E.Hecherman, John S.breese, and Eric.J.Horvitz (1989), The complication of decision models, In Uncertainty in Artificial Intelligence 5, pages 162-173.
- [12] P.E.Lehner, and A.Sadigh (1993), Two procedures for compiling influence diagrams, In Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, pages 335-341.
- [13] J. F. Nash (1951), Non-cooperative games, Annals of Mathematics.
- [14] Lemke and Howson (1964), Equilibrium points in bimatrix games, J. Society of Applied Mathematics, 12(2): 413-423.
- [15] Scarf, H., and T. Hansen (1973), The Computation of Economic Equilibrium. New Haven: Yale University Press.
- [16] Jie Bao (2003), Notes on Graphical Game, In <http://www.cs.iastate.edu/~baojie>, Dept of Computer Science, Iowa State University.



Zhao Yun received the BEng. Degree from computer science department of YunNan University (CSYNU) in 2001. Currently, she is a master student of CSYNU. Her research interests are related to Computational Game Theory and Multi-agent System, data and knowledge engineering. She has published several conference and journal papers in these areas.



Liu Wei-yi graduated from Huazhong University of Science and Techn ology in 1976. He is a professor of the School of Information Science and Engineering of YunNan University. He is a member of IEEE Computer Society. His research interests include fuzzy systems, data and knowledge engineering.

Genetic Searching for Optimized Closure State of CFST Arch Bridge Construction

Fan Jian-Feng¹, Zhong Luo², Tong Qi-Wei²

¹Civil Engineering and Architecture School,

²Computer Science and Technology School, Wuhan University of Technology, Wuhan 430070, the P.R. China

Email: fjeff@mail.whut.edu.cn Tel: +86 (0)27- 87658008 Fax: +86 (0)27- 87211983

ABSTRACT

It is very difficult to find the initial state of the backward analysis in a traditional simulative calculation of bridge construction because of the bearing peculiarity of the Concrete Filled of Steel Tubular (CFST) arch bridge. In this paper, the genetic algorithm had to search for the optimized closure state in the arch-rib construction, which can be taken as the initial state for the backward analysis. According to the detailed analysis, the objective function was determined, which included the adjustment increment of cable length as the variable parameter. But the objective function could only be obtained by the structural simulation calculation, which makes the searching more complicated. Hence a method using floating-point genetic algorithm was adopted and the genetic operation was adjusted to improve the precision and speed of the genetic search. Finally the optimized closure state in arch-rib construction was obtained with an improved genetic algorithm in an example calculation. It was shown that the results of the method could be used as the initial state for the backward analysis.

Keywords: CFST Arch Bridge, Simulative Calculation of Construction, Backward Analysis, Genetic Algorithm, Floating-point Coding

In recent years, more attention has been paid to the Concrete Filled of Steel Tubular Arch Bridge which has developed rapidly in China due to the advantages of its bearing capability and its method of construction. But the bigger the span, the more difficult will be the construction. Hence, a simulative calculation of construction should be applied to control the structure during construction, to determine the proper state in every phase, and to confirm that the final state fits the design. A simulation calculation of the construction has been the main method for calculating the state of modern bridge structure^[1].

1 . INTRODUCTION

The simulation calculation of construction mainly includes forward and backward analysis. By different types of bridge construction techniques, many experts and scholars have improved and advanced it. Considering in particular the CFST arch bridge in bearing capability and construction techniques, the research and development of a simulation calculation of the construction is necessary. The traditional backward analysis is not applicable to the construction calculation of arch bridge, because it is very difficult to find out the initial state of the backward analysis in the traditional simulation calculation of bridge construction^[2], which can be taken as the final state of an arch bridge in construction. The final state of the arch bridge is made up of 2 parts: the

final state of the whole bridge and the closure state of the main arch-circle. When the arch-rib is closed, it is difficult to optimize the line type and bearing force. Therefore, it is very important to search the optimized closure state in arch-rib construction, which can not be taken only as the optimized objective of the state before closure in the arch-rib construction, but also the initial state for the backward analysis. Then we can optimize the state of every step in the arch-rib hoisting construction. However, it is very hard to obtain the optimized closure state of arch rib with the traditional calculation method.

The Genetic Algorithm, actively used over recent years, is a capable tool for solving global optimization problems. Problems in the constraints, which lack a clear analytic relationship, can be solved by the Genetic Algorithm. Furthermore, a GA can easily find the global optimal solution by a stochastic adaptation search algorithm. In this paper, the Genetic Algorithm and Simulation calculation are combined to search for the optimized closure state in the arch-rib construction.

2. PROCESS OF THE SOLUTION BY GENETIC ALGORITHM

2.1 Main Process

When we apply the GA to the objective function of the closure state in construction, the inputting parameter should be confirmed firstly. The optimal solution of the objective function is the optimal closure state, for which we search. In real construction, commonly, if the line type of the arch-rib offset from the design line type, the cable should be adjusted to meet the design. Adjusting the cable can also decrease the danger strain of the arch rib. Thus the whole steel tubular arch-rib will be safe in an equal bearing state. Many literatures provide many methods to optimize the line type of the bridge and obtain the rational state by adjusting the cable force. For a steel tubular arch bridge, we think that the adjustment of cable forces result from the change of the internal force, so that the whole cable force will vary with the adjustment of one cable. Moreover, the precision of an actual testing force is not high for the construction error. So the Generic Algorithm searching process will be affected strongly if cable force is taken as a parameter, for which there is no clear analytic relationship.

We know that, when a cable is adjusted, the cable is elongated by the jack, and the part of the cable stretched out will stay beyond the anchorage bearing without participating in the suspension, so the length of the adjusted cable will virtually be shortened. Since it is very easy to adjust the cable length on the construction site, we can take the flex length, which is along the vector of cable axis as the parameter $\{x\}$. Thus the problem of searching for the

optimal state in construction can be solved by searching for a population of optimized flex quantities of cable adjustments to obtain the optimal line type of the arch-rib for the closure state in arch-rib construction by a Genetic Algorithm.

The mathematical model for structural state of bridge is described by Eq. (1):

$$[K] \{\mu\} = \{F\} + \{\Delta F\} \quad (1)$$

where, $[K]$, $\{\mu\}$ and $\{F\}$ are the stiffness matrix, displacements vector and forces vector respectively, and $\{\Delta F\}$ denotes the equivalent node force caused by the flex quantity of cable adjustment. As research in literature^[2] shows, $\{\Delta F\}$ can be taken as the equivalent node force caused by the flex quantity of cable adjustment which is $\{x\}$ for a non-force reason (such as the decrease of temperature), as in Eq. (2):

$$\{\Delta F\} = \{\Delta F(x_i)\} \quad (2)$$

Hence, when a population of $\{x\}$ is given, a structural state of steel tubular arch bridge is obtained.

2.2 Confirm Objective Function

The objective function should be confirmed by using a Genetic Algorithm to obtain the optimal closure state in arch-rib construction. The optimal solution of the objective function is the optimal closure state for which we search.

As far as the arch bridge is concerned, the key of obtaining the optimal state of the arch bridge is to confirm the line type of the arch-rib match the design so that the cable force and steel stress meet the constraints in intension, distortion, stability when the steel arch-rib is integrated into the objective function in Eq. (3):

$$\min f(\{x\}) = \sum_i [\mu_i(\{x\})]^2 \quad (3)$$

Where $\mu_i(\{x\})$ denotes the displacement of the arch-rib line type, which can be provided by Eq. (1).

As research shows in the literature^[2], constraints for the parameter should be as shown below:

1) Cable forces: all cable forces should be tensioned, and are defined as: $F_i(\{x\}) \geq 0$.

2) Stress in the position of splicing: The section of splicing should always be pressed, and it must be less than allowable stress, namely: $0 \leq g_k(\{x\}) \leq \overline{g_k}$

3) The stress of the most dangerous section of the arch-rib must be less than the allowable maximum press as follows:

$$h_m(\{x\}) \leq \overline{h_m}$$

Considering the Generic Algorithm as a kind of unconstrained optimization method, the penalty function strategy is needed to transform the constrained problem to an unconstrained problem. Thus, the objective function is transformed to a new one as shown below:

$$\begin{aligned} \min f(\{x\}) = & \sum_i [\mu_i(\{x\})]^2 + \\ & (C \times t)^\alpha \left[\sum_i (\max \{0, -F_i\})^\beta + \right. \\ & \left. \sum_k (\max \{0, g_k - \overline{g_k}\})^\beta + \sum_m (\max \{0, h_m - \overline{h_m}\})^\beta \right] \end{aligned}$$

in which t denotes generations of the evolution,

$C=0.5, \alpha=\beta=2$. With the increasing of generations, the penalty pressure of a non-feasible solution will grow rapidly as well.

2.3 Genetic Algorithm and Process

The Objective function, with the cable adjustment amount as a parameter, is not a clear mathematic formula, so a structural calculation should be performed to obtain the objective function. Considering the complex fitness function evaluation and the control variable' dimension, an appropriate genetic algorithm should be adapted to improve the genetic search and to shorten the generations of evolution. The length of chromosome will be long if it's coded by using the binary system. Thus, the searching space of Genetic Algorithm will expand rapidly, and the convergence speed will reduce rapidly as well. So we can use the method of Floating-point Coding, which is no the encoding and decoding operation. And it is spontaneous for continuous parameter optimization. One real parameter vector correspond one chromosome, while one real variable corresponds to one gene. The Genetic Algorithm, which applied to the floating-point coding gene, can gradually change during the search process. Genetic searching can be improved as follows:

1) Initially, the population $P(0)$ and the genes of chromosome are generated random in (x_j^b, x_j^t) to meet the design:

$$x_j = x_j^b + \beta_j(x_j^t - x_j^b)$$

Where, $j=1 \sim n$, n denotes the chromosome length of coding.

2) the fitness function population $P(t)$ should be put into formula (1) to simulate the state of construction, An evaluation of the fitness function should be made, and the result should be ranked according to their values.

3) The reproduction and crossover $P^1(t)$ should be reproduced and selected from the population $P(t)$ by the ranking, so that the population $P^2(t)$ will be obtained by reproducing $P^1(t)$, according to the crossover probability, the k and m chromosome gene x_k^j and x_m^j must be selected randomly from the population $P^2(t)$, then proceed to crossover, after which the offspring chromosome gene is as follows:

$$\begin{aligned} x'_{kj} &= \lambda_j x_{kj} + (1 - \lambda_j) x_{mj} \\ x'_{mj} &= \lambda_j x_{mj} + (1 - \lambda_j) x_{kj} \quad (j=1 \sim n) \end{aligned}$$

Where, λ_j denotes a random real number in $[0, 1]$, which ensures that $x_{kj} + x_{lj} = x'_{kj} + x'_{lj}$ before crossover or after it. In this way, the population $P^3(t)$ is obtained.

4) The chromosome of population is mutated on the mutation probability. Variable of chromosome x_{ij} is calculated as follows:

$$\text{When } 0 < \zeta \leq 0.5, \quad x''_{ij} = x_{ij} - (x_{ij} - x_j^b)\eta;$$

$$\text{When } 0.5 < \zeta < 1, \quad x''_{ij} = x_{ij} - (x_{ij} - x_j^t)\eta$$

Where x''_{ij} denotes the variable value after mutation.

A random number ζ enable x_{ij} to add or subtract by jumping, the jumping value is controlled by η .

5) Since the computation of the fitness function is complex, three maneuvers can be used to obtain the termination

of genetic algorithm search process. One maneuver is to set the largest number of generations evolution (200 is chosen in this the example below). Another maneuver is to set the continuous unchangeable generation of the optimal individual (such as the fifth generation). The third one is to set a precision limit so that searching will terminate when the precision is the limit.

3. EXAMPLES AND ANALYSIS

A simulative calculation of construction, which embedded by the GA, is applied to the example of the *NanNiDu* arch bridge. The calculated brief drawing for the state before the closure of the *NanNiDu* arch bridge is shown in *Figure 1*, and character parameters are shown in *Table 1*.

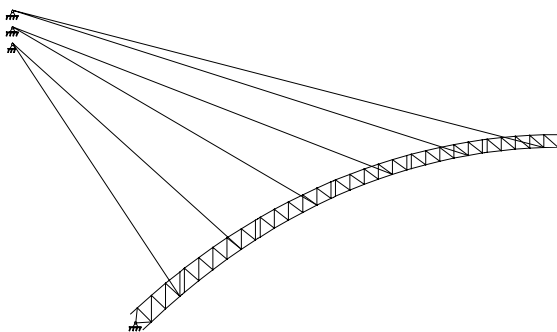


Figure 1 the simple chart of calculation for *NanNiDu* arch bridge

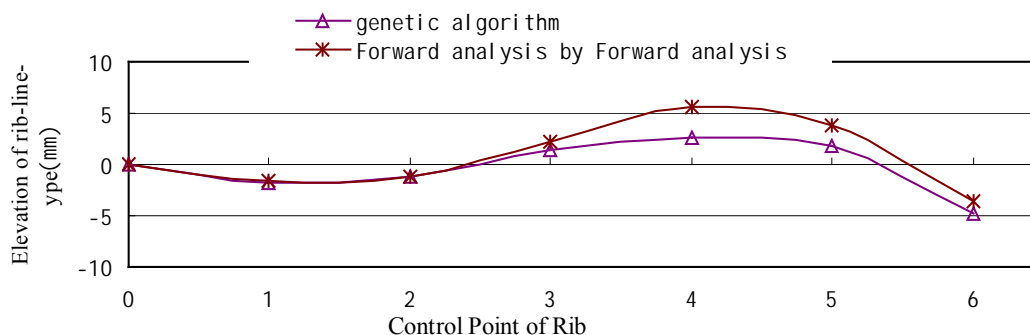


Figure 2 the line type comparison prior to closure of arch-rib

Table 2 the comparison of reasonable state

Content	Relative elevation of line type (mm)						Cable force (kN)					
	1	2	3	4	5	6	1	2	3	4	5	6
Genetic Algorithm	-1.8	-1.2	1.5	2.6	1.8	-4.7	21.4	32.1	200	191	656	961
Forward Analysis	-1.6	-1.1	2.3	5.6	3.8	-3.5	93.4	27.1	90.4	339.1	618.3	944.8

4. CONCLUSIONS

In this paper, a Genetic Algorithm has been applied to the Simulative Calculation of Construction to obtain the optimized closure state in arch-rib construction, which can

Table 1 parameters of structural calculation

Name	Main arch	Belly pole	Cable
E (GPa)	206	206	190
EI (kN*m ²)	1.69E6	5.44E4	1.71
EA (kN)	1.65E7	3.60E6	2.05E5

The symmetry of the structure is taken into account and the *Ernst formula* is applied to amend the hold down effect of the cables. As shown in *Figure 1*, there are 6 suit cables, so the parameter variable is set as 6 dimensions. For a population size is 30, the mutation probability is 4%~8%. By changing some of the genetic parameters during the search process, the optimized solution is obtained at the 108th generation. Thus the cable force can be obtained by calculation. Computation results provided by the genetic algorithm and the forward analysis method based on iterative theory are shown in *Figure 2* and *Table 2*.

As shown in the chart and table, the line type of the optimized closure state calculated by the genetic algorithm and forward analysis method based on iterative theory are approximate. Most forces are almost the same except 1th, 3th, 4th group, because the internal force is redistributed due to the later elevation difference of the arch-rib segment. And the results by the GA search are better.

Once the optimized closure state is obtained, the optimized state of every state in the construction can be calculated by backward analysis. Taking into account the situation where the results by forward analysis and backward analysis don't match, results should be adjusted to make the curve in *chart 2* into one curve.

be taken as the initial state for backward analysis. Hence, the optimized state of every state in the construction can be calculated and the construction can be controlled well. Given the real line type and cable force of the optimized closure state in an arch-rib construction, we can search for the

optimal adjustment for the length of cable, so that the final line type of the arch rib fits the design.

More researches should be done, such as how to calculate the optimized state of every state in the construction by a Genetic Algorithm and the application of a Parallel Compute in the Simulative Calculation of Construction.

5. ACKNOWLEDGEMENT

Thanks to Hubei Province Traffic Science Program and Wuhan University of Technology 2003th Fund Program for their assistance with this research.

6. REFERENCES

- [1] Xiao Rucheng, Study on theory and method of determining structural reasonable state for CFST [D]. Shanghai: doctoral degree paper of Tong Ji University, 1996. (In Chinese)
- [2] Yuan Haiqing, Fan Xiaochun, Fan Jianfeng, Zhou Qiangxin, "Iterative go-ahead method of rib-assembling predicting for long-span concrete filled steel tubular arch bridge", China Journal of Highway and Transport, Vol 16, No. 3, October 2003, pp. 48~51. (In Chinese)
- [3] Chen Dewei, Bai Zhizhou, Huang Zheng, "Determination of Initial Cable Forces for Prestressed Concrete Cable-stayed Bridges with Genetical Algorithm", Journal of Tong Ji University, Vol. 31, No. 1, 2003, PP. 11~15. (In Chinese)
- [4] Zhang Jianmin, Zheng Jielian, Qing Rong, "Adjustment calculation of Buckle-cable Forces during segmental construction of CFST [J]", Journal of China & Foreign Highway, Vol. 22, No. 4, 2002, PP. 44~47. (In Chinese)
- [5] Li Mingqiang, Kou Jisong, Ling Dan, Li Shuquan, The basic theory and method of genetic algorithm [M]. Beijing: science press, 2002. (In Chinese)



Fan Jian-feng is a doctoral student in Civil Engineering from Wuhan University of Technology. His research interest is in intelligent technology applied to civil engineering.



Zhong Luo is a Full Professor. He graduated from Wuhan University in 1982; His research interests are in intelligent technology, software engineering, and image graphic.

A Wavelength Assignment Algorithm of Parallel LU Decomposition Communication Pattern on WDM Ring Interconnection Network

Yawen Chen, Fangai Liu

Department of Computer Science, Shandong Normal University,
Jinan, Shandong, China, 250014

E-mail: chyw1980@hotmail.com

ABSTRACT

Wavelength assignment is a key topic in WDM optical interconnection networks. Since there are different communication patterns according to different parallel algorithms, how to realize these communication patterns on optical interconnection networks is a hot research field. Based on the WDM ring interconnection network, a kind of parallel LU decomposition communication pattern is designed and the wavelength assignment of realizing this communication pattern on WDM ring is discussed. By embedding the communication pattern of a special bipartite graph into the WDM ring, an algorithm to embed the parallel LU decomposition communication pattern into the WDM ring is designed. The minimum number of wavelengths required to realize this communication pattern on WDM ring with n^2 nodes is $n^2/4$ when n is even and $(n^2-3)/4$ when n is odd.

Keywords: LU decomposition, wavelength assignment, parallel processing, WDM ring, network embedding

1. INTRODUCTION

The solution of linear systems is a computational bottleneck in many scientific computing problems and LU decomposition^[1] is important for many scientific applications, which is usually performed in parallel form in order to improve its computing speed. With the increasing computation power of parallel computers, interprocessor communication has become an important factor that limits the performance of supercomputing systems. Optical interconnection networks, whose advantages have been well demonstrated on wide and local area networks, are promising networks for future supercomputers. With the development of WDM^[2] technology, we can take full advantage of the parallel transmission characteristic of WDM optical interconnections to embed the complex communication patterns into the simple optical interconnections. Thus, the network topologies can be considerably simplified.

As we know, wavelength assignment is a key topic in WDM optical interconnection networks. How to make full use of the optical wavelength channels efficiently has attracted a lot of attention. Optimal routing and channel assignments for hypercube communication on optical mesh-like processor arrays are studied in [2]. In [3], we give a wavelength assignment algorithm of hypercube communication pattern on optical RP(k) networks and improve the result in [2]. We also give some results about the parallel FFT communication pattern on a class of regular optical interconnection networks in [4].

The ring topology is widely used in interconnection networks and the implementation of WDM ring is regarded as the first

deployment phase of optical networking. In this paper, we discuss the wavelength assignment of parallel LU decomposition communication pattern on WDM ring. Firstly, we construct the parallel LU decomposition communication pattern, and then by embedding the communication pattern of a special bipartite graph SBG (p) into WDM ring, we design an algorithm to embed the parallel LU decomposition communication pattern into WDM ring. We identify the lower bound of the wavelength required and design an embedding way named bipartite interval mapping which can achieve the lower bound. Finally, we derive the minimum numbers of wavelengths needed to realize the parallel LU decomposition on the WDM ring with n^2 nodes is $n^2/4$ when n is even and $(n^2-3)/4$ when n is odd.

2. PRILIMINARIES

2.1. WDM optical interconnection

Wavelength Division Multiplexing (WDM) divides the bandwidth of an optical fiber into multiple wavelength channels, so that multiple users can transmit at distinct wavelength channels through the same fiber concurrently. To efficiently utilize the bandwidth resources and to eliminate the high cost and bottleneck caused by optoelectronic conversion and processing at intermediate nodes, end-to-end lightpaths are usually set up between each pair of source-destination nodes. A connection or a light path in a WDM network is an ordered pair of nodes (x,y) corresponding to that a packet is sent from source x to destination y. There are two approaches for establishing a connection in a network whose links are multiplexed with virtual channels^[5]. One is called Path Multiplexing (PM), in which the same channel has to be used on each link along a path, and the other is called Link Multiplexing (LM), in which different channels may be used. In this paper, we assume that no wavelength converter facility is available in the network. Thus, a connection must use the same wavelength throughout its path. In this case, the lightpath satisfies the wavelength-continuity constraint.

2.2. Wavelength Assignment

Routing and channel assignment (RCA^[2]) tries to minimize the number of channels to realize a communication requirement by taking into consideration both routing options and channel assignment options. The RCA problem can be described as follows. Given a set of all-optical connections, the problem is to (a) find routes from the source nodes to their respective destinations, and (b) assign channels to these routes so that the same channel is assigned to all the links of a particular route. The goal of RCA is to minimize the number of assigned channels.

Many researchers have studied the RCA problem in the WDM networks^[2-5]. This paper considers optimal routing and channel assignment (RCA) schemes to realize a kind of parallel LU decomposition communication pattern on WDM

ring interconnection network.

2. PLU COMMUNICATION PATTERN

In many scientific computational applications, the LU decomposition of matrix is the most common approach in solving the linear equations. Any non-singular matrix A can be expressed as a product $A = LU$, there exist exactly one lower triangular matrix L and exactly one upper triangular matrix U.

To solve the matrix equation $Ax = (LU)x = L(Ux) = b$, first solve $Ly = b$ for y. Then solve $Ux = y$ for x. In this process, LU decomposition is the most important step.

$$\text{We assume } A = \begin{bmatrix} a_{11} & \Lambda & a_{1n} \\ M & & M \\ a_{n1} & \Lambda & a_{nn} \end{bmatrix}, L = \begin{bmatrix} l_{11} & \Lambda & l_{1n} \\ M & & M \\ l_{n1} & \Lambda & l_{nn} \end{bmatrix},$$

$$U = \begin{bmatrix} u_{11} & \Lambda & u_{1n} \\ M & & M \\ u_{n1} & \Lambda & u_{nn} \end{bmatrix}.$$

The formulas for the computation of L and U are

$$a_{ij} = \sum_{k=1}^n l_{ik} u_{kj} = \begin{cases} \sum_{k=1}^{i-1} l_{ik} u_{kj} + u_{ij}, & i \leq j \\ \sum_{k=1}^{i-1} l_{ik} u_{kj} + l_{ij} u_{ij}, & i > j \end{cases} \Rightarrow$$

$$\begin{cases} u_{1j} = a_{1j}, 1 \leq j \leq n, u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}, 2 \leq i \leq j \\ l_{i1} = a_{i1} / u_{11}, i = 2, \dots, n, l_{ij} = (a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}) / u_{jj}, i > j > 2 \end{cases}$$

In Fortran 77 syntax, the algorithm (without pivoting), is coded as follows:

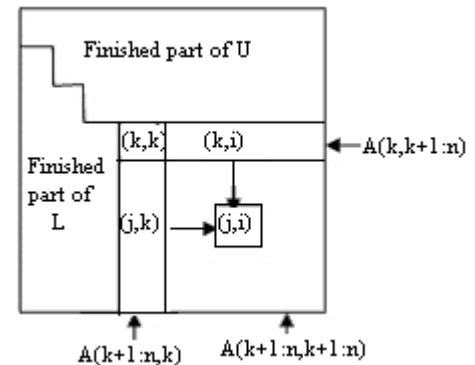
```
DO k = 1, n-1
  DO x = k+1, n
    A(x, k) = A(x, k) / A(k, k)
    !Column Normalization
  END DO
  DO i = k+1, n
    DO j = k+1, n
      A(i, j) = A(i, j) - A(i, k)*A(k, j)
      !Submatrix
    END DO
  END DO
END DO
```

It can be seen that the codes have the parallel nature. For the parallel LU decomposition, we first consider the problem of data allocation on the processors during the implement of parallel LU decomposition. The block cyclic data distribution^[1] of matrix is commonly used in partitioning matrix. Given $E = P \times Q$ processors and $M \times N$ matrix A_{MN} , the data blocks of the matrix whose size is $r \times s$ are distributed cyclically onto the $P \times Q$ processors. If element $A(i, j)$ is allocated onto the processor $E(p, q)$, then $p = (i/s) \bmod P$ and $q = (j/t) \bmod Q$. For example, given $P \times Q = 4 \times 4 = 16$ processors, each processor is identified

by (p, q) uniquely. Fig.1. (a) illustrates the block cyclic data distribution with $M \times N = 10 \times 10$ and $r \times s = 1 \times 3$ on the processors mentioned above. The notation on the data element stands for the processor on which the data is allocated.

0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,2	0,3	0,3
1,0	1,0	1,0	1,1	1,1	1,1	1,2	1,2	1,3	1,3
2,0	2,0	2,0	2,1	2,1	2,1	2,2	2,2	2,3	2,3
3,0	3,0	3,0	3,1	3,1	3,1	3,2	3,2	3,3	3,3
0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,2	0,3	0,3
1,0	1,0	1,0	1,1	1,1	1,1	1,2	1,2	1,3	1,3
2,0	2,0	2,0	2,1	2,1	2,1	2,2	2,2	2,3	2,3
3,0	3,0	3,0	3,1	3,1	3,1	3,2	3,2	3,3	3,3
0,0	0,0	0,0	0,1	0,1	0,1	0,2	0,2	0,3	0,3
1,0	1,0	1,0	1,1	1,1	1,1	1,2	1,2	1,3	1,3

(a) Block Cyclic Data Distribution



(b) The kth Step of LU Decomposition

Fig.1 LU Decomposition

In the kth step of LU decomposition, communications take place between processor S and D if the block in which A_{ik} or A_{kj} is allocated is assigned to processor S and the block in which A_{ij} is allocated is assigned to processor D. Otherwise, communications will not take place between S and D. Fig.1.(b) illustrates the communications in the kth step of LU decomposition.

Based on the block cyclic data distribution, we consider the processors in the LU decomposition as the nodes of the communication pattern and the communications as the edges. The interprocessor communication pattern in each step during the LU decomposition has the same characteristics. Fig.2.(a) illustrates the communication pattern of the kth step of LU decomposition.

For the sake of simplicity, we assume that the large scale of matrix is allocated onto $n \times n = n^2$ ($n \geq 2$) processors by the block cyclic data distribution and each processor is identified by (i, j) uniquely, where $0 \leq i, j \leq n-1$. Let m_0 denote processor $(0,0)$, c_i denote processor $(0,i)$ and v_{ij} denote processor (i,j) . Then the parallel LU decomposition communication pattern can be denoted by graph $G_n(V_n, E_n)$ and the set of nodes V_n and the set of edges E_n can be denoted by the following forms:

$$V_n = \bigcup_{i=1}^{n-1} \{c_i, r_i\} \cup \bigcup_{i=1}^{n-1} \bigcup_{j=1}^{n-1} \{v_{ij}\} \cup \{m_0\},$$

$$E_n = \left(\bigcup_{i=1}^{n-1} \{(m_0, c_i), (m_0, r_i)\} \right) \bigcup_{i=1}^{n-1} \bigcup_{j=1}^{n-1} \{(r_i, v_{ij}), (c_j, v_{ij})\}.$$

For simplicity in notation, we use PLU denote this parallel LU decomposition communication pattern in the following context.

PLU has the following properties:

Property 1: The edges (m_0, c_i) , (c_i, v_{ii}) , (r_i, v_{ii}) and (m_0, r_i) in G_n construct a cycle when their directions are ignored. The total number of such cycles is $n-1$. ($1 \leq i \leq n-1$)

Property 2: The degree of node v_{ij} is 2. ($1 \leq i, j \leq n-1$)

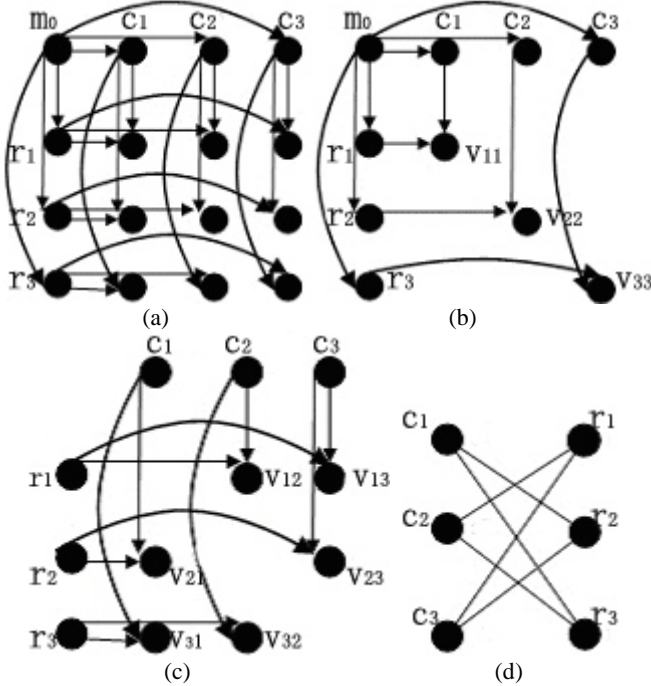


Fig.2 PLU and its decomposition and simplification (n=4)

3. WAVELENGTH ASSIGNMENT OF PLU ON WDM RING

We model a network as a graph $G(V, E)$, where nodes in V are switches and edges in E are links. We assume that the nodes of the ring are ordered counterclockwise from 1 to $2p$, and the edges of the ring are ordered counterclockwise from 1 to $2p$ denoted by $e_i (1 \leq i \leq 2p)$. In general, an optical WDM network consists of routing nodes interconnected by point-to-point fiber links, which can support a certain number of wavelengths. In this paper, we ignore the directions of the networks for the sake of simplicity, that is $(x, y) = (y, x)$. The routing and wavelength assignment are subject to the following two constraints:

1. Wavelength continuity constraint: a lightpath must use the same wavelength on all the links long its path from source to destination edge node.
2. Distinct wavelength constraint: all lightpaths using the same link must be allocated distinct wavelengths.

4.1 Decomposition of PLU

The PLU communication pattern $G_n(V_n, E_n)$ can be decomposed into two sub graph $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$,

$$V_1 = \bigcup_{i=1}^{n-1} \{c_i, r_i\} \bigcup_{i=1}^{n-1} \{m_0\}$$

$$E_1 = \bigcup_{i=1}^{n-1} \{(m_0, c_i), (m_0, r_i)\} \bigcup_{i=1}^{n-1} \{(r_i, v_{ii}), (c_i, v_{ii})\}$$

$$V_2 = \bigcup_{i=1}^{n-1} \{c_i, r_i\} \bigcup_{i=1}^{n-1} \bigcup_{j=1}^{n-1} \{v_{ij}\}$$

$$E_2 = \bigcup_{i=1}^{n-1} \bigcup_{j=1}^{n-1} \{(r_i, v_{ij}), (c_j, v_{ij})\}$$

$$E_1 \cap E_2 = \emptyset, E_1 \cup E_2 = E_n.$$

Fig. 2.(b) and (c) are the sub graphs of PLU when n is equal to 4.

Lemma 1: Realizing G_1 on WDM ring requires $n-1$ wavelengths.

Proof: According to property 1, the edges (m_0, c_i) , (c_i, v_{ii}) , (r_i, v_{ii}) and (m_0, r_i) in G_n construct a cycle when their directions are ignored. The total number of such cycles is $n-1$. The number of wavelengths required to realize these four lightpaths is 1 if and only if these four lightpaths do not share the same link and they cover all the links in the ring. Therefore, realizing G_1 on WDM ring needs $n-1$ wavelengths.

4.2 Simplification of PLU

According to property 2, we construct the homeomorphism graph $G_3(V_3, E_3)$ for sub graph G_2 . That is to say, for the two edges connecting v_{ij} , remove the node v_{ij} with degree of 2 and contract the two edges into one edge. Fig. 2.(d) is the simplification graph of G_2 when $n=4$. It is obvious that graph $G_3(V_3, E_3)$ is just a bipartite graph. We assume that $p=n-1$ ($p \geq 1$) and the nodes in V_3 are divided into two node sets

which are denoted by $C = \bigcup_{i=1}^p \{c_i\}$ and $R = \bigcup_{i=1}^p \{r_i\}$.

The edge set of G_3 can be denoted by

$$E_3 = \bigcup_{i=1}^p \bigcup_{j=1}^p \{(r_i, c_j)\} - \bigcup_{i=1}^p \{(r_i, c_i)\} \quad \text{or} \quad E_3 = \bigcup_{i=1}^p \bigcup_{j=1, j \neq i}^p \{(r_i, c_j)\}.$$

For the sake of simplicity, the communication pattern with $2p$ nodes of this special bipartite graph is denoted by SBG(p).

Lemma 2: The minimum number of wavelengths required to embed SBG(p) into WDM ring is equal to that required to embed graph G_2 into WDM ring.

We can prove that the number of wavelengths required to embed SBG(p) into WDM ring is no less than that required to embed graph G_2 into WDM ring. When the lightpaths (r_i, v_{ij}) and (c_j, v_{ij}) in G_2 do not share the same physical link and they are assigned the same wavelength, the number of wavelengths required to embed SBG(p) into WDM ring is equal to that required to embed graph G_2 into WDM ring.

After the decomposition and simplification of PLU, we give the following result:

Theorem 1: The minimum number of wavelengths to realize PLU on WDM ring $W_{PLU}(n) = W_{SBG}(n-1) + (n-1)$, where $W_{SBG}(n-1)$ denotes the minimum number of wavelengths to realize SBG(p) on WDM ring.

Proof: From Lemma 1 we know that if we assign the wavelengths in the set of $\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_{n-1}\}$ for the $n-1$ cycles, then these wavelengths will no longer be used by the lightpaths in G_2 . From Lemma 2 we know that the minimum

number of wavelengths required to embed SBG (p) into WDM ring is equal to that required to embed graph G_2 into WDM ring. Therefore, $W_{PLU}(n) = W_{G_2}(n-1) + (n-1)$.

Thus, we can get the minimum number of wavelengths to realize PLU on WDM ring if we obtain the minimum number of wavelengths to realize SBG (p) on WDM ring.

4.3 A Lower Bound of SBG (p) Embedded in WDM Ring

In this section, we first introduce an approach to get the lower bound of wavelengths in the optical networks:

Remove some links from the physical topology. Thus divide the original topology into two sub graphs of G_a and G_b . Let the removed links construct the set of U, which is called link cuts. For the lightpaths to establish in the optical topology, only the lightpath whose two ends are not in the same sub graph would transpose the links in U. We call these lightpaths cut channels D_u . Suppose they are distributed evenly in the links of U, then the Maximum number of lightpaths each link contains

$$\text{is } \left\lceil \frac{|D_u|}{|U|} \right\rceil.$$

The number of wavelengths required to realize any communication patterns must satisfy $W \geq \max_{\forall U} \left\lceil \frac{|D_u|}{|U|} \right\rceil$.

As we know, the size of any cuts of a ring topology is 2, that is to say, $|U|=2$. Then the number of wavelengths required realizing any communication patterns on WDM ring must

$$\text{satisfy } W \geq \left\lceil \frac{\max_{\forall U} \{|D_u|\}}{2} \right\rceil.$$

Using the approach mentioned above to obtain the lower bound of wavelengths, we give the following result:

Theorem 2: The number of wavelengths required to realize SBG (p) on WDM ring with $2p$ nodes, denoted by $w(p)$,

$$\text{satisfies } w(p) \geq \begin{cases} \left(\frac{p-1}{2}\right)^2, & p \text{ is even.} \\ \frac{p^2}{4} - \frac{p}{2} + 1, & p \text{ is odd.} \end{cases}$$

Proof: We select the link cuts of $U(S, T, i) = \{e_i, e_{(i+p) \bmod 2p}\}$, where i is an integer between 1 and $2p$. More specifically, e_i and $e_{(i+p) \bmod 2p}$ is the farthest edges in the ring. Such link cuts divide the ring into two sub graphs S and T which have the same number of nodes. We map the nodes of C and R in SBG (p) onto the nodes of WDM ring. Suppose the number of nodes in C mapping onto the sub graph S is m , then the number of nodes in C mapping onto the sub graph T is $p-m$. The number of nodes in R mapping onto the sub graph S is $p-m$ and the number of nodes in R mapping onto the sub graph T is m . Thus, the number of lightpaths transposing the edges e_i and $e_{(i+p) \bmod 2p}$ in the link cuts of $U(S, T, i)$, which we denoted by $U(m)$, can be calculated by the following formula:

$$\begin{aligned} U(m) &= m^2 + (p-m)^2 - \Delta \geq m^2 + (p-m)^2 - p \\ &= 2\left(m - \frac{p}{2}\right)^2 + \frac{p^2}{2} - p \end{aligned} \quad (1)$$

Where $0 \leq m \leq p$.

$$\Delta = \left| \{i \mid c_i \in S \text{ and } r_i \in T\} \right| + \left| \{i \mid c_i \in T \text{ and } r_i \in S\} \right| \text{ and } \Delta \leq p.$$

$\Delta = p$ if and only if there do not exist $i = j$ that make $c_i \in S, r_j \in S$ and $c_i \in T, r_j \in T$, that is to say, c_i and r_i do not belong to the same set of S and T .

Now, we discuss the formula (1).

(1) p is odd: When $m = (p-1)/2$ or $(p+1)/2$, and $\Delta = p$, $U(m)$ obtains its minimum value of $p^2/2 - p + 1/2$, which is denoted by $B(p)$. We assume that $U(e_i, e_j)$ represents the number of lightpaths transposing link e_i and e_j , then

$$\max_{e_i, e_j \in E} (U(e_i, e_j)) \geq B(p).$$

(2) p is even: When $m = p/2$ and $\Delta = p$, $U(m)$ obtains its minimum value of $p^2/2 - p$, which is denoted by $A(p)$. We can prove by the reduction to absurdity that there do not exist the mapping from C and R to the nodes of the ring which can make the number of lightpaths transposing e_i and $e_{(i+p) \bmod 2p}$ achieve the minimum value of $A(p)$. Thus,

$$\max_{e_i, e_j \in E} (U(e_i, e_j)) \geq A(p) + 1.$$

According to (1) and (2), we know that the number of wavelengths required to realize SBG(p) on WDM ring, denoted by $w(p)$, satisfies

$$\begin{aligned} w(p) &\geq \left\lceil \frac{\max_{e_i, e_j \in E} (U(e_i, e_j))}{2} \right\rceil \\ &\geq \begin{cases} \left\lceil \frac{B(p)}{2} \right\rceil = \left(\frac{p-1}{2}\right)^2, & p \text{ is odd.} \\ \left\lceil \frac{A(p)+1}{2} \right\rceil = \frac{p^2}{4} - \frac{p}{2} + 1, & p \text{ is even.} \end{cases} \end{aligned}$$

4.4 SBG (p) embedded in WDM ring

In order to embed SBG (p) into the WDM ring, we should first establish the mapping from the nodes of SBG (p) to the nodes of the ring topology. We define an arrangement of the nodes in SBG (p) which is denoted by X_n as follows.

When p is odd, the arrangement of the nodes X_1, \dots, X_{2p} is $c_1, r_2, \dots, r_{p-1}, c_p, r_1, c_2, \dots, c_{p-1}, r_p$.

When p is even, the arrangement of the nodes X_1, \dots, X_{2p} is $c_1, r_2, \dots, c_{p-1}, r_p, r_1, c_2, \dots, r_{p-1}, c_p$.

If we map the i th node of X_n onto the i th processor of the WDM ring topology, thus we establish the 1-1 mapping from the nodes of SBG(p) to the nodes of WDM ring. In addition, the lightpaths of SBG(p) are routed by the shortest path whether in the clockwise or counterclockwise directions. We call this embedding way bipartite interval mapping.

We have the following result in the WDM ring by the bipartite interval mapping.

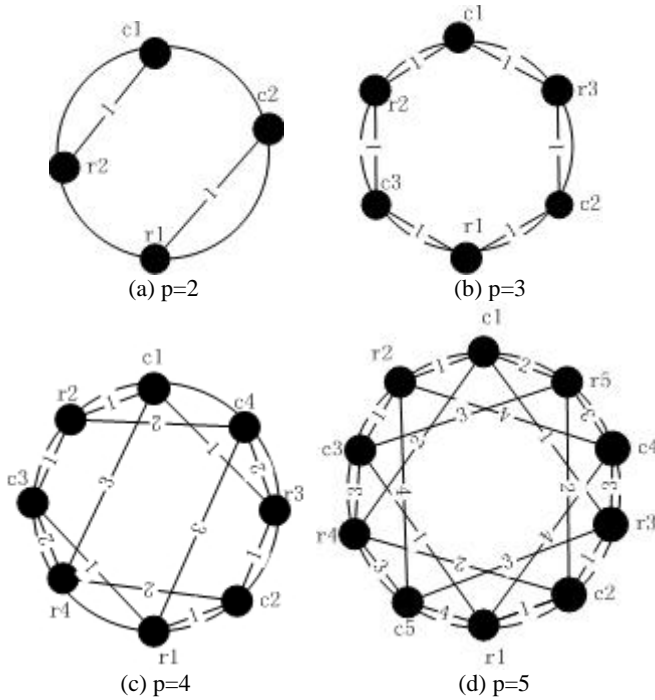


Fig.3 The embedding of SBG(p) into WDM ring

Theorem 3: By the embedding way of bipartite interval mapping, the number of wavelengths required to realize SBG(p) with $2p$ nodes is

$$(a) \text{ when } p \text{ is even, } w(p) = \frac{p^2}{4} - \frac{p}{2} + 1 \quad (2)$$

$$(b) \text{ when } p \text{ is odd, } w(p) = \left(\frac{p-1}{2}\right)^2 \quad (3)$$

Proof: We prove this problem by the mathematical induction.

(1) We can prove that the formulas are true for $p=2$ and 3 , as we can see from Fig.3 (a) and (b).

(2) We suppose the formulas are true for $p \leq k$, then we discuss for $p = k+1$ when k is odd and even respectively.

(a) When k is odd and $k+1$ is even, the sets of nodes and edges in SBG(p) can be expressed in the flowing forms:

$$V(k+1) = V(k) \cup \{r_{k+1}, c_{k+1}\},$$

$$E(k+1) = E(k) \cup \left\{ \bigcup_{j=1}^k \{(r_{k+1}, c_j), (c_{k+1}, r_j)\} \right\}.$$

For $p=k+1$, the additional lightpaths in comparison with $p=k$

$$\text{is } \Delta E(k+1) = \bigcup_{j=1}^k \{(r_{k+1}, c_j), (c_{k+1}, r_j)\}.$$

With these additional lightpaths routed in the shortest path, we denote the additional wavelengths required to realize these additional lightpaths on WDM ring by $\Delta w(k+1)$.

As we can see, lightpaths (c_{k+1}, r_k) , (c_{k+1}, r_{k-1}) , (r_{k+1}, c_k) and (r_{k+1}, c_{k-1}) routed in the shortest path do not share any physical link, so that we can assign the same wavelength to these lightpaths.

The wavelength assignment scheme is stated as follows.

(1) Assign wavelength $\lambda_{(j/2)+1}$ to (c_{k+1}, r_{k-j}) , $(c_{k+1}, r_{k-(j+1)})$, (r_{k+1}, c_{k-j}) and $(r_{k+1}, c_{k-(j+1)})$, where $j = 0, 2, \dots, k-3$;

(2) Assign wavelength $\lambda_{(k+1)/2}$ to lightpaths (c_{k+1}, r_1) and (r_{k+1}, c_1) .

For example, lightpaths (c_{k+1}, r_k) , (c_{k+1}, r_{k-1}) , (r_{k+1}, c_k) and (r_{k+1}, c_{k-1}) is assigned wavelength λ_1 ; Lightpaths (c_{k+1}, r_3) , (c_{k+1}, r_2) , (r_{k+1}, c_3) and (r_{k+1}, c_2) is assigned wavelength $\lambda_{(k-1)/2}$.

Thus, the set of wavelengths assigned to the lightpaths in $\Delta E(k+1)$ is $\Delta \lambda_{k+1} = \{\lambda_1, \dots, \lambda_{(k-1)/2}, \lambda_{(k+1)/2}\}$. Then,

$$\Delta w(k+1) = |\Delta \lambda_{k+1}| = (k+1)/2.$$

According the assumption, the number of wavelengths required to realize the lightpaths in $E(k)$ is $w(k) = \left(\frac{k+1}{2}\right)^2$.

Therefore, the number of wavelengths required to realize SBG(p), denoted by $w(k+1)$, can be calculated as follows:

$$w(k+1) = w(k) + \Delta w(k+1) = \frac{(k+1)^2}{4} - \frac{k+1}{2} + 1.$$

Therefore, formula (2) is true.

From the above discussion, we can see that the result is true when $k+1$ is even.

Next, we will prove that the result is true when $k+1$ is odd.

(b) When k is even and $k+1$ is odd, the sets of nodes and edges in SBG(p) can be expressed in the flowing forms:

$$V(k+1) = V(k) \cup \{r_k, c_k, r_{k+1}, c_{k+1}\},$$

$$E(k+1) = E(k) \cup \Delta E(k) \cup \Delta E(k+1),$$

$$\Delta E(k) = \bigcup_{j=1}^{k-1} \{(r_k, c_j), (c_k, r_j)\},$$

$$\Delta E(k+1) = \bigcup_{j=1}^k \{(r_{k+1}, c_j), (c_{k+1}, r_j)\}.$$

Similar to the above analysis, we assign wavelengths for the lightpaths in $\Delta E(k)$ and $\Delta E(k+1)$ and denote the number of wavelengths required to realize these lightpaths by $\Delta w(k, k+1)$.

The wavelength assignment scheme is stated as follows:

(1) Assign wavelength $\lambda_{(j+1)/2}$ to lightpaths (r_{k+1}, c_j) ,

(c_j, r_k) , (r_k, c_{j+1}) and (c_{j+1}, r_{k+1}) , where $j = 1, 3, \dots, k-3$;

(2) Assign wavelength $\lambda_{k/2}$ to lightpaths (r_{k+1}, c_{k-1}) , (c_{k-1}, r_k) , (r_k, c_{k+1}) , (c_{k+1}, r_{k-1}) , (r_{k-1}, c_k) and (c_k, r_{k+1}) ;

(3) Assign wavelength $\lambda_{k/2 + (j+1)/2}$ to lightpaths (c_{k+1}, r_j) , (r_j, c_k) , (c_k, r_{j+1}) and (r_{j+1}, c_{k+1}) , where $j = 1, 3, \dots, k-3$.

Therefore, the set of wavelengths required to realize the lightpaths in $\Delta E(k)$ and $\Delta E(k+1)$ is

$$\Delta \lambda_{k, k+1} = \{\lambda_1, \dots, \lambda_{(k/2)-1}, \lambda_{k/2}, \lambda_{k/2+1}, \dots, \lambda_{k-1}\}.$$

$$\text{Thus, } \Delta w(k, k+1) = |\Delta \lambda_{k, k+1}| = k-1.$$

According the assumption, the number of wavelengths required to realize the lightpaths in $E(k-1)$ is $w(k-1)$. Thus, the number of wavelengths required to realize SBG(p), denoted by $w(k+1)$, can be calculated as follows:

$$w(k+1) = w(k-1) + \Delta w(k, k+1) = \left(\frac{(k+1)-1}{2}\right)^2.$$

Therefore, the result is true when $k+1$ is odd.

From (a) and (b) we know that the results hold for $p=k+1$.

Fig.3.(c) and (d) illustrate the embedding of SBG(p) into WDM ring when $p=4$ and $p=5$. The notations on the lines represent the wavelengths assigned to the according lightpaths.

From the above discussion, we know that the lower bound to embed SBG(p) in WDM ring is given and an embedding way to achieve the lower bound is designed during the proving

of Theorem 3. Thus, we get the following result:

Theorem 4: The minimum number of wavelengths required to realize SBG(p) on WDM ring, denoted by $w_{\min}(p)$,

$$\text{satisfies } w_{\min}(p) = \begin{cases} (\frac{p-1}{2})^2, & p \text{ is odd.} \\ \frac{p^2}{4} - \frac{p}{2} + 1, & p \text{ is even.} \end{cases}$$

4.5 The embedding of PLU in WDM ring

Theorem 5: The minimum number of wavelengths required to realize PLU on WDM ring with n^2 ($n=p+1$) nodes satisfies

$$W_{LU}(n) = \begin{cases} \frac{n^2}{4}, & n \text{ is even} \\ \frac{n^2+3}{4}, & n \text{ is odd} \end{cases}$$

Proof: Using Theorem 1 and Theorem 4, when n is even and p

$$\text{is odd, } W_{LU}(n) = (\frac{(n-1)-1}{2})^2 + (n-1) = \frac{n^2}{4}.$$

$$\text{When } n \text{ is odd and } p \text{ is even, } W_{LU}(n) = \frac{n^2+3}{4}.$$

5. CONCLUSION

In this paper, we proposed a wavelength assignment algorithm of a kind of parallel LU decomposition communication pattern on WDM ring interconnection network, which requires $n^2/4$ wavelengths when n is even and $(n^2-3)/4$ when n is odd. Our result can be used to implement the LU decomposition on WDM ring, which is important in the theory and practice.

Since there are different communication patterns according to different parallel algorithms, how to realize these communication patterns on optical interconnection networks is a hot research field. More research on RCA considering various parallel communication patterns for parallel algorithms may be a worthwhile effort.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No: 60373063) and the Natural Science Foundation of Shandong (No: Y2002G03).

6. REFERENCES

- [1] Kai Shen, Tao Yang et al, "S+: Efficient 2D Sparse LU Factorization on Parallel Machines". In SIAM Journal on Matrix Analysis and Applications (SIMAX), Vol. 22, No.1, 2000, pp. 282-305.
- [2] Yuan X, Melhem R, "Optimal routing and channel assignments for hypercube communication on optical mesh-like processor arrays", In: Johnsson SL, ed. Proceedings of the 5th International Conference on Massively Parallel Processing Using Optical Interconnection. Las Vegas, NV: IEEE Press, 1998, pp. 110-118.
- [3] Fangai Liu, ZY Liu, XZ Qiao, "A wavelength assignment algorithm of hypercube communication on optical RP(k) networks", Journal of Software, Vol.14, No.3, 2003, pp. 282-305. (In Chinese)

[4] Fangai Liu, Yawen Chen, "Wavelength Assignment of Parallel FFT Communication Pattern in a Class of Regular Optimal WDM Network," Proceedings of The IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN 2004), pp. 495-500, May 10-12, 2004, Hong Kong.

[5] H. Zang, JP Jue, and B. Mukherjee, "A Review of Routing and Wavelength Assignment Approaches for Wavelength-Routed Optical WDM Networks", Optical Networks Magazine, Vol. 1, No. 1, Jan. 2000, pp. 47-60.

[6] Jonathan Gross, Jay Yellen, Graph Theory and its Applications, Boca Raton: CRC Press, 1999.

Chen Yawen is a postgraduate student of Shandong Normal University pursuing her master's degree. Research interests include parallel computing, interconnection networks and optical interconnection.

Liu Fangai is a Professor of Shandong Normal University. He graduated from the Chinese Academy of Science. He is a Supervisor of Doctor and his research interests include parallel computing, interconnection networks and computer applications.

Genetic Algorithms for Solving Graphical Games*

Jin Li Wei-yi Liu Yun Zhao

Computer Science Department, YunNan University
Kunming, Yunnan (650091), China
Email: bluecube@sina.com Tel.: 13888824219

ABSTRACT

Finding equilibria is a core task for graphical games. A genetic algorithm is presented for computing an exact equilibrium of graphical games with arbitrary graphical structure through exploiting structural properties of graphical games. Our algorithm has capability of global optimization and converges to a Nash equilibrium with much more probability than previous approach. Experiment results show our algorithm can find a high-quality Nash equilibrium in much larger games.

Keywords: Genetic Algorithms, Graphical Games, Nash equilibrium

1. INTRODUCTION

Game theory is a mathematical framework that describes interactions between multiple rational agents and allows for reasoning about their outcomes. However, the complexity of standard game descriptions grows exponentially with the number of agents involved. For many multi-agent situations, this blowup presents a serious problem. Recent work in artificial intelligence^[1-3] proposes the use of structure game representations that utilize a notion of locality of interaction; these representations allow a wide range of complex games to be represented compactly.

In this paper we consider the task of computing a *Nash equilibrium* for *Graphical Games*. A Nash equilibrium is a strategy profile such that it is no agent's interest to deviate unilaterally. A naive approach to finding Nash equilibria is to convert the structured game into a standard game representation, and apply a standard game-theoretic solution algorithm^[6]. This approach is, in general, infeasible for all but the simplest games. Kearns^[1] provide an algorithm for finding equilibria of graphical games. But Kearns' methods only apply to the tree structured graphical games and solving a relaxed problem: finding an approximate Nash equilibrium. Vickrey^[5] propose a greedy hill-climbing approach which finding approximate equilibria for graphical games with arbitrary structure. But Vickrey's algorithm is not guaranteed to find a global optimal solution, in other words the hill-climbing algorithm does not always converges to Nash equilibrium.

Genetic algorithms have demonstrated considerable success in provide good solutions to many complex optimization problems. They have been well documented by numerous pieces of literature, such as^{[7][8]}. In this paper we present a

genetic algorithm for solving graphical games. Our approach apply to find exact equilibria. It has global optimization capacity and converges to an equilibrium with much more probability than the previous approach. We finally provide some preliminary experimental results demonstrating our algorithm can find high-quality Nash equilibria in much larger games.

2. GRAPHICAL GAMES

In this section, we introduce some basic notation and terminology for game theory and then describe the framework of graphical games.

2.1 Game Theory

The conceptually simplest representation of game is the *normal form*^[6]. A n -player, k -action normal game is defined by a set of n matrices M_i ($1 \leq i \leq n$). Each player ($Agent_i$) choose an action s_{ij} from its action set $S_i = \{s_{i1}, s_{i2}, \dots, s_{ik_i}\}$. For simplicity of notation, we assume that $k_1 = k_2 = \dots = k_n = k$. The entry $M_i(s_1, s_2, \dots, s_n) = M_i(s)$ specifies the payoff to $Agent_i$ when the joint action of the n agents is $s \in S_1 \times S_2 \times \dots \times S_n$. Thus, each M_i has k^n entries. The action s_{ij} is the *pure strategy* of each $Agent_i$. The agents are also allowed to play a *mixed strategies* δ_i . A mixed strategy δ_i is a probability distribution over pure strategies. $\delta_i : S_i \rightarrow [0,1]$ assigns probability $\delta_i(s_{ij})$ to action s_{ij} ,

where $\sum_{j=1}^k \delta_i(s_{ij}) = 1$. Each $Agent_i$'s mixed strategy is

statistically independent of those of his opponents. The joint mixed strategies of the n agents is $\sigma = \langle \delta_1, \delta_2, \dots, \delta_n \rangle$, where δ_i is denoted a discretionary mixed strategy of $Agent_i$. σ is called a mixed strategy profile of game. We use (σ_{-i}, δ'_i) to denote the vector which is the same as σ except the i^{th} component, where the value has been changed to δ'_i . Given a mixed strategy profile σ , we define the notion of an expected payoff that $Agent_i$ can receive from σ as:

Definition 1 If $s = (s_1, s_2, \dots, s_n)$ is a pure strategy profile, where s_i denote a discretionary pure strategy of $Agent_i$. $\sigma = \langle \delta_1, \delta_2, \dots, \delta_n \rangle$ is a mixed strategy profile, δ_i denote a discretionary mixed strategy of $Agent_i$. $S = S_1 \times S_2 \times \dots \times S_n$ denote the space of pure strategy profiles, with element s , thus

$$u_i(\sigma) = \sum_{s \in S} [\delta_1(s_1) \delta_2(s_2) \dots \delta_n(s_n)] u_i(s),$$

where $u_i(s)$ can be directly get from M_i .

*This work is supported by the National Natural Science Foundation of China (Grant No.60263003), the Foundation of the Key Laboratory of Intelligent Information Processing, Institute of computing technology, Chinese Academy of Sciences (Grant No.IIP 2002-2) and the Yunnan Natural Science Foundation (Grant No.2002F0011M).

Note that $Agent_i$'s expected payoff to a mixed strategy profile is a linear function of $Agent_i$'s mixing probability δ_i , a fact which has many important implications.

	L	M	R
U	4,3	5,1	6,2
M	2,1	8,4	3,6
D	3,0	9,6	2,8

Figure 1 : Payoff matrix of two agents

For instance, in figure 1 a mixed strategy for $Agent_1$ is a vector: $(\delta_1(U), \delta_1(M), \delta_1(D))$ such that $\delta_1(U), \delta_1(M)$ and $\delta_1(D)$ are nonnegative and $\delta_1(U) + \delta_1(M) + \delta_1(D) = 1$. δ_2 is a $Agent_2$'s mixed strategy. The $Agent_1$'s expected payoff to profile $\sigma = (\delta_1, \delta_2)$, where $\delta_1 = (1/3, 1/3, 1/3)$ and $\delta_2 = (0, 1/2, 1/2)$, is:

$$\begin{aligned} u_1(\delta_1, \delta_2) \\ = 1/3(0 \times 4 + 1/2 \times 5 + 1/2 \times 6) + 1/3(0 \times 2 + 1/2 \times 8 + 1/2 \times 3) + 1/3 \\ (0 \times 3 + 1/2 \times 9 + 1/2 \times 2) = 11/2. \end{aligned}$$

Similarly, $u_2(\delta_1, \delta_2) = 27/6$.

A Nash Equilibrium is a profile of strategies such that each agent's strategy is an optimal response to the other agents' strategies. In other words, no agent can improve their expected payoff by deviating unilaterally from a Nash Equilibrium strategy profile. Before we introduce the definition of Nash Equilibrium, we first give the definition of *regret degree* of $Agent_i$ with respect to a mixed strategy profile σ . (Note a pure strategy profile is a degenerated mixed strategy profile)

Definition 2 Given a mixed strategies profile σ , we define the *regret degree* of $Agent_i$ with respect to σ to be the most $Agent_i$ can gain (on expectation) by diverging from the strategy profile σ :

$$Reg_i(\sigma) = \max_{\delta_i'} (u_i(\sigma_{-i}, \delta_i') - u_i(\sigma))$$

If $Agent_i$ can not improve their expected payoff by deviating unilaterally from σ , it will not change its' strategy in σ . Thus, we get $Reg_i(\sigma) \geq 0$.

Now, we can use the definition of *regret degree* to define the notation "Nash Equilibrium".

Definition 3 A mixed-strategy profile σ^* is a *Nash Equilibrium* if, for all $Agent_i$, $Reg_i(\sigma) = 0$. σ^* is an \mathcal{E} -*Nash Equilibrium* if, for all $Agent_i$, $Reg_i(\sigma) \leq \mathcal{E}$.

Someone will ask the question: "Given a normal form Game, whether a Nash Equilibrium of the Game must be existed?" The following theorem 1 answers this question.

Theorem 1^[6] There is at least a mixed-strategy Nash Equilibrium for every finite normal form Game.

2.2 Graphical Games

The size of the payoff arrays required to describe a normal-form game grows exponentially with the number of agents. Kearns^[1] introduced the framework of graphical game. Graphical games capture local structure in multi-agent interactions, allowing a compact representation for scenarios where each agent's payoff is only affected by a small subset of other agents. Examples of interactions where this structure occurs include agents that interact along organization hierarchies and agents that interact according to geographic proximity.

Example 1: Consider the following example, based on a similar example in^[2]. Suppose a road is being built from north to south through undeveloped land, and $2n$ agents have purchased plots of land along the road. The agents w_1, w_2, \dots, w_n on the west side and the agents e_1, e_2, \dots, e_n on the east side. Each agent needs to choose what to build on its land: a factory, a shopping mall, or a residential complex. Its payoff depends on what he builds and on what is built north, south, and across the road from its land. All of the decisions are made simultaneously. We can represent this case as the following graphical games in figure 2 (the payoff matrices of agents are omitted):

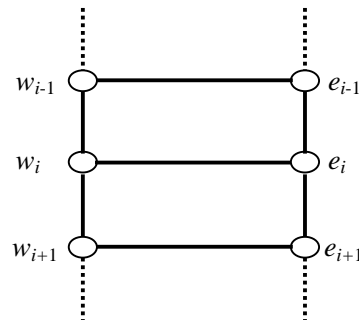


Figure 2. the Road graphical games

(The circles represent the agents and the line between the circles represents the agents' payoff is relevant to each other.)

An n -agent graphical game is a pair $\langle Gr, M \rangle$, where Gr is an undirected graph on n vertices and M is a set of n matrices M_i ($1 \leq i \leq n$), called the local game matrices. $Agent_i$ is represented by a vertex labeled i in Gr . We use $N_{Gr}(i) \subseteq \{1, \dots, n\}$ to denote the set of neighbors of $Agent_i$ in Gr . By convention, $N_{Gr}(i)$ always includes i itself. The interpretation is that each agent is in a game with only their neighbors in Gr . Thus, if $|N_{Gr}(i)| = k$ ($k \leq n$), the matrix M_i has k indices, one for each agent in $N_{Gr}(i)$. Note that the normal form representation of the example 1 consists of $2n$ matrices each of size 3^{2n} , whereas in the graphical games, each matrix has size at most 3^4 . if $s \in S_{i1} \times S_{i2} \times \dots \times S_{ik}$ ($i, i2, \dots, ik \in N_{Gr}(i)$), $M_i(s)$ denotes the payoff to i when his k neighbors (which include himself) play s . The expected payoff under a mixed strategy profile σ is defined analogously.

Note that our definitions are entirely representational, and alter nothing about the underlying game theory. Thus, every graphical game has a Nash Equilibrium.

3. A GENETIC ALGORITHM FOR SOLVING GRAPHICAL GAMES

In this section, we define a fitness function that measures the distance of a given strategy profile away from a Nash equilibrium. we then design a genetic algorithm that starts from a random initial strategy profile (individual) set and a new set of approximations is created by the process of the selecting individuals according to their level of fitness and reproducing them using variation operations. The algorithm gradually improves the profiles until a Nash Equilibrium is reached.

More precisely, for a strategy profile σ , we define $Sreg(\sigma)$ to be the sum of the regrets of the agents:

$$Sreg(\sigma) = \sum_{i=1}^n Reg_i(\sigma)$$

This function is nonnegative and is equal to 0 exactly when σ is a Nash Equilibrium.

Now we discuss the computation problem of $Sreg(\sigma)$ in graphical games. Given a profile σ , $u_i(\sigma)$ in definition1 is a constant. According to the definition1, $u_i(\sigma_{-i}, \delta_i')$ can be looked as k -dimension linear function of δ_i' . So $Reg_i(\sigma)$, the $Agent_i$'s regret degree, is easily solved by the following optimization problem:

$$\begin{aligned} & \text{Maximize: } \mathbf{C}\mathbf{x} \\ & \text{Subject to: } \sum_{j=1}^k x_{ij} = 1 \\ & \quad \forall j \quad 0 \leq x_{ij} \leq 1 \end{aligned}$$

where $\mathbf{C} = (c_1, c_2, \dots, c_k)$ and $\mathbf{x} = (x_{i1}, x_{i2}, \dots, x_{ik})$ is probabilistic vector. In above optimization problem, the parameters are strategy probabilities of $Agent_i$, and the coefficients involve the payoffs only of i and its neighbors. Thus, the optimization problem can be solved efficiently based on the given strategy profile σ , the $Agent_i$'s payoff matrix and that of its neighbors in the graph.

3.1 Representation Structure

Here, we use a vector $\sigma = \langle \delta_1, \delta_2, \dots, \delta_n \rangle$ as a chromosome, where δ_i is a probabilistic vector:

$$\delta_i = \langle x_{i1}, x_{i2}, \dots, x_{ik} \rangle$$

($\sum_{j=1}^k x_{ij} = 1$ and $0 \leq x_{ij} \leq 1$). So the representation structure of

the algorithm is:

$$\langle x_{11}, x_{12}, \dots, x_{1k}, x_{21}, x_{22}, \dots, x_{2k}, \dots, x_{n1}, x_{n2}, \dots, x_{nk} \rangle$$

We define an integer s as the number of chromosomes and initialize s chromosomes randomly.

3.2 Evaluation Function

Evaluation function, denoted by $eval(\sigma)$, is to assign a probability of reproduction to each chromosome σ so that its likelihood of being selected is proportional to its fitness relative to the other chromosome in the population, that is,

the chromosome with higher fitness will have more chance to produce offspring by using *roulette wheel selection*.

Let $\sigma_1, \sigma_2, \dots, \sigma_s$ be the s -size chromosomes at the current generation. According to the objective function values, the s -size chromosomes can be rearranged from good to bad, i.e. the better the chromosome is, the smaller ordinal number it has. Now let a parameter $a \in (0,1)$ in genetic algorithm be given, then we can define the so-called *rank-bank evaluation function* as follows:

$$eval(\sigma_i) = a(1-a)^{i-1}, i = 1, 2, \dots, s.$$

We mention that $i=1$ means the best individual, $i=s$ means the worst individual

3.3 Selection Process

The selection process is based on spinning the roulette wheel s times, each time we select a single chromosome for a new population in the following way.

Step1. Calculate the cumulative probability p_i for each chromosome σ_i :

$$p_0 = 0 \quad p_i = \sum_{j=1}^i eval(\sigma_j), i = 1, 2, \dots, s.$$

Step2. Generate a random real number r in $(0, p_s]$.

Step3. Select the i^{th} chromosome σ_i such that $p_{i-1} < r < p_i$.

Step4. Repeat the Step2 and Step3 s times and obtain s copies of chromosomes.

3.4 Crossover Operation

We define a parameter P_c of a genetic algorithm as the probability of crossover. In order to determine the parents for crossover operations, let us do the following process repeatedly from $i=1$ to s : generating a random real number r from the interval $(0,1)$, the chromosome σ_i is selected as parent if $r < P_c$. We denote the selected parents as $\sigma'_1, \sigma'_2, \sigma'_3, \dots$ and divide them to the following pairs:

$$(\sigma'_1, \sigma'_2), (\sigma'_3, \sigma'_4), (\sigma'_5, \sigma'_6), \dots$$

Let us illustrate the crossover operator on each pair by (σ'_1, σ'_2) . At first, we generate a random number c from the interval $(0,1)$, then the crossover operator on σ'_1, σ'_2 will produce two children X and Y as follows:

$$X = c \cdot \sigma'_1 + (1-c) \cdot \sigma'_2 \quad \text{and}$$

$$Y = (1-c) \cdot \sigma'_1 + c \cdot \sigma'_2.$$

Because the feasible set is convex, this arithmetical crossover operation ensures that both children are feasible if both parents are

3.5 Mutation Operation

We define a parameter P_m of a genetic system as the probability of mutation. Similar to the process of selecting parents for crossover operation, we repeat the following steps from $i=1$ to s : generating a random real number r from the interval $(0,1)$, the chromosome σ_i is selected as a parent for mutation if $r < P_m$. For each selected parent, denoted by $\sigma_i = \langle \delta_1, \delta_2, \dots, \delta_n \rangle$, we mutate it by the following way. We randomly select δ_j from σ_i in term of uniform

distribution and mutate it to a new random probabilistic vector δ'_j . Thus we get a mutated chromosome σ'_i .

3.6 The Procedure

Following selection, crossover and mutation, the new population is ready for its next evaluation. The genetic algorithm will terminate after a given number of cyclic repetitions of the above steps. We can summarize the genetic algorithm for solving the Nash Equilibrium of graphical games as follows.

- Step0. Input parameters a, s, P_c, P_m .
- Step1. Initialize s -size chromosomes.
- Step2. Update the chromosomes by crossover and mutation operations.
- Step3. Update the chromosomes by crossover and mutation operations.
- Step4. Calculate the objective values for all the chromosomes.
- Step5. Compute the fitness of each chromosome by rank-based evaluation function based on the objective values.
- Step6. Select the chromosome by spinning the roulette wheel.
- Step7. Repeat the Step2 to Step6.
- Step8. Report the best chromosome as the optimal solution.

4. EXPERIMENT RESULTS

We compared our results to the published results of the hill-climbing algorithm of Vickrey and Koller^[5] (VK hereafter).

	rock	paper	scissors
rock	0,0	+1,-1	-1,+1
paper	-1,+1	0,0	+1,-1
scissors	+1,-1	-1,+1	0,0

Figure 3 : Payoff matrix of rock-paper-scissors

Following VK, our algorithm was run on Road game (see the example 1) of varying size. We constructed a game where the payoff for an agent is simply the sum of payoffs of games played separately with its neighbors, and where each agent subgame has the payoff structure of the rock-paper-scissors payoff matrix of figure3.

For our genetic algorithm and VK algorithm, we chose a set of game sizes to run on. For each game size, we solve the Road game with both genetic algorithm and VK algorithm. Genetic algorithm started with the following parameters: the population size is 20 (the algorithm started with 20 random probabilistic vectors), the probability of crossover P_c is 0.3, the probability of mutation P_m is 0.2, the parameter a in the rank-based evaluation function is 0.05. The running time and equilibrium error of experiments are shown in figure 4

Figure 4 (a) shows the running time of two algorithms in varying game size: solid line for genetic algorithm, dotted line for VK hill-climbing. We can see that VK hill-climbing runs faster than genetic algorithm in varying size game. As the size of games is larger, genetic algorithm becomes costly. Figure 4 (b) shows the equilibrium quality of two algorithms.

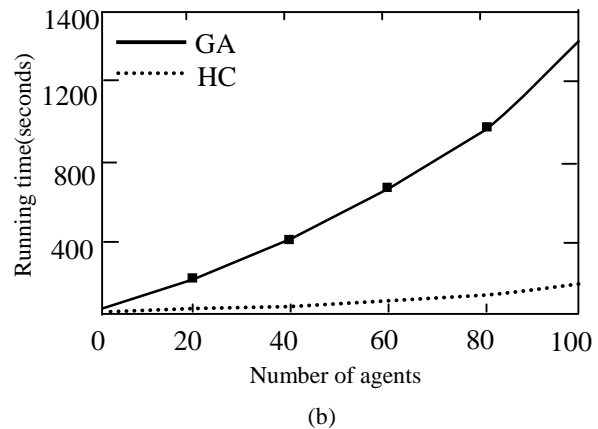
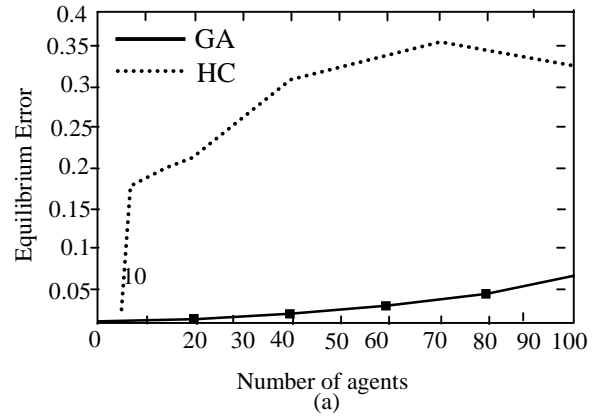


Figure 4 running time and equilibrium error of experiments

We can see that although hill-climbing runs more efficiently, the equilibrium error of it also is unacceptable. The errors of the equilibria produced by hill-climbing grow with the game size, a consequence of the fact that the hill-climbing search is over a higher dimensional space. On the other side, our genetic algorithm has capability of global optimization and the experiments results indicate that our genetic algorithm produces much better equilibrium than hill-climbing method.

5. CONCLUSIONS

In the last few years, several papers have addressed the issue of finding equilibria in structured games. In this paper, We present a genetic algorithm for computing a Nash equilibrium in Graphical Games. Our algorithm has capability of global optimization and there is much more probability of converging to an Nash equilibrium. We showed that our techniques provide much better solutions for graphical games with very large number of agents than previous approach.

6. REFERENCES

- [1] M. Kearns, M. Littman, and S. Singh Graphical models for game theory. *In Proc. UAI2001*.
- [2] D. Koller, B. Milch. Multi-agent influence diagram for representing and solving games. *In Proc. 17th IJCAI-01*, 1027 -1034, 2001.

- [3] P. La Mura. Game networks. *In Proc. UAI2000*, pages 335-342, 2000.
- [4] M. L. Littman, M. Kearns and S. Singh. An efficient exact algorithm for singly connected graphical games. *In NIPS-14*, pages 817-823, 2002.
- [5] D. Vickrey and D. Koller. Multi-agent algorithms for solving graphical games. *In Proc. AAAI*, 2002.
- [6] R. D. McKelvey and A. McLennan. Computation of equilibrium in finite games. In *Handbook of computational Economics*, vol.1, pages 87-142. Elsevier, 1996.
- [7] D. Fudenberg, and J. Tirole. *Game Theory*. MIT Press, 1991.
- [8] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [9] Z. Michalewicz, *Genetic Algorithms+Data Structures= Evolution Programs*, 2nd edition, Springer-Verlag, New York, 1994.

Some Issues on Adaptive Genetic Algorithm

Zhang Jianjian, Wang Pan
School of Automation, Wuhan University of Technology
Wuhan, 430070, P.R. China
Email: jfpwang@tom.com Tel: 86-27-87858435

ABSTRACT

One of the important branches of genetic algorithm (GA)——adaptive GA has been discussed. Based on the introduction of adaptation with different level(s) and several kinds of formalized descriptions of adaptive GA, three forms of adaptation have been analyzed. Several mathematical expressions of adaptive operators are discussed and the corresponding schema theorems of adaptive GA with linear and nonlinear expressions are studied.

Keywords: Adaptation, Genetic Algorithms, Adaptive Genetic Operators, Population Scale, Schema Theorem.

1. INTRODUCTION

Genetic algorithm (GA) is a kind of random searching algorithm that simulates the mechanism of natural selection and inheritance, which was first presented by Prof. Holland of Michigan University in 1975[1]. For the intelligent property of effective robustness, the ability of global optimization, parallel problem-solving and so on, GA has drawn great attention and been developed in theory and application in the past 10 years [2,3]. Now, it has become one of the important approaches of soft computing together with some other technique simulating the laws of evolution in nature.

In this paper, some key issues of the important branch of GA——adaptive genetic algorithm (AGA), such as the form of adaptation, the comments on several existed mathematical expressions of adaptive operators, and schema theorem of CA with adaptive operators are discussed.

2. SEVERAL DESCRIPTIVE FORMS

During the growing and reproduction process of lives, individuals themselves adapt the organic structure of to strengthen the fitness to the environment. This is named as adaptation in biology. Inspired by such property, in the bionic algorithms including GA, we think it's also always be required that the algorithms should adjust their internal structures or sub-algorithms according to the interaction and feedback information from the environment. So adaptation can be thought as one of the vital characteristics. Adaptive GA has the different levels including environment, population, individual and gene at least, which correspond with different structures and different adaptive parameters. Xu Zongben *et al.* have given the formulized description as follows [2]:

$$GA(e_t, J_t, S_t, C_t, M_t, \Sigma t) \quad (1)$$

Where: e_t —the coding format in the t -th generation of AGA (including the coding length, the decoding rule);

J_t —the measure of fitness in the t -th generation of AGA;

S_t —the selecting operation of choosing in the t -th generation of AGA;

C_t —the crossover operation in the t -th generation of AGA (including the crossover probability $P_c(t)$);

M_t —the mutation operation in the t -th generation of AGA (including the mutation probability $P_m(t)$);

Σ —systemic parameters in the t -th generation of AGA (such as the ending criterion, the population size N_t , the immigrating probability I_t of the individuals, etc).

Because of the configuration of multi-population of AGA, it can be described more intensively as:

$$AGA(e_t^i, J_t^i, S_t^i, C_t^i, M_t^i, \Sigma t^i, i_t) \quad (2)$$

Where, it denotes the i -th population in the t -th generation.

Following, a classical adaptive mode will be discussed in the levels of environment, population and individual.

2.1 Adaptive Fitness Evaluation Function

As the bionic algorithm simulating the biology system, evolution algorithm has to face the problems whether the result could fit its variable space (the space of the result) during its running. The problem of fitness is one of the keys of the evolution algorithm in a way, and it decides the direction of the algorithm——all the other adjustments (selection, crossover, mutation, immigration, etc) are operated on the base of evaluation for the variable space (the environment) for the results. Therefore, the construction of the fitness function is a very important task. In the literature [6], the author describes detailedly the construction of the basic fitness function in order to solve the problems of the adaptation of the fitness function in dynamic environment.

2.2 Adaptive Population Size

The size of the population of the simple GA is a fixed value, that is to say it keeps invariable from the start to the end during the running process. In the beginning of the running, the larger population makes for raising the searching multiplicity of the algorithm, however, the results tend to convergence and the fitness of the individuals tend to stable value(s) with the running, here, if we still keep the original population scale, it's obvious the cost of the algorithm will rise instead of improving the searching efficiency. Thereby, it's necessary and significant to make the populations size adaptive during the running of the algorithm in time. The adaptation of the population's size will be more complex and more frequent for multi-population because it is necessary to make the individuals immigrate and syncretize adaptively among these populations for the situation of the other populations' running.

2.3 Adaptive Genetic Operators

In the nature with the law--- “individuals compete and the nature elects, the fit test for survive”, the individuals with better fitness get more survival opportunities because of the weaker selecting pressure. In the applications of GA, the biologic characteristic is also been utilized, that is to say: to protect the individual with higher fitness, specifically with high surviving probability for selection operator and the lower probability for crossover and mutation; in the other hand, opposite strategies are done for the individual with lower fitness. In the later discussion, we will give more details on the adaptive operators.

3. SEVERAL FORMS OF THE ADAPTIVE OPERATORS

Following is the relative notations: P_c -the crossover probability, P_m -the mutation probability, f_{max} -the maximum fitness of the population, f_{min} -the minimum fitness of the population, f' -the larger fitness between the two individuals, \bar{f} -the average fitness of one epoch of the population, f_i -the fitness for the individual i .

From literatures, it can be found some researchers have done some work on such issue. For example, Chen Changzheng *et al.* proposed the following operators [4]:

$$P_c = k_1 \frac{f_{max} - f'}{f_{max} - \bar{f}} \quad (3)$$

$$P_m = k_2 \frac{f_{max} - f_i}{f_{max} - \bar{f}} \quad (4)$$

where , $k_1, k_2 \leq 1.0$

However, there are some mistakes in these operators (for instance, the mutation operator): if $f_i < \bar{f}$, $\frac{f_{max} - f_i}{f_{max} - \bar{f}} > 1$,

with f_i approaching f_{min} , the value of $\frac{f_{max} - f_i}{f_{max} - \bar{f}}$ will approach

infinity. Therefore, although $k_2 \leq 1.0$, the value of $\frac{f_{max} - f_i}{f_{max} - \bar{f}}$

is still uncontrollable, and it's difficult to adapt the value of the parameter k_2 . Thereby, the authors propose the following to avoid the former problems:

$$P_c = C_0 + (C_1 - C_0) \frac{f_{max} - f'}{f_{max} - f_{min}} \quad (5)$$

$$P_m = C_0' + (C_1' - C_0') \frac{f_{max} - f_i}{f_{max} - f_{min}} \quad (6)$$

where, $C_0 < C_1$, $C_0' < C_1'$ and $C_0, C_1, C_0', C_1' \in [0,1]$.

Meanwhile, the operators with the piecewise form Srinivas etc. proposed can also avoid the former problems, as follows [5]:

$$P_c = \begin{cases} P_{c_1} - (P_{c_1} - P_{c_2}) \frac{f' - \bar{f}}{f_{max} - \bar{f}}, & f' \geq \bar{f} \\ P_{c_1}, & f' < \bar{f} \end{cases} \quad (7)$$

$$P_m = \begin{cases} P_{m_1} - (P_{m_1} - P_{m_2}) \frac{f_i - \bar{f}}{f_{max} - \bar{f}}, & f_i \geq \bar{f} \\ P_{m_1}, & f_i < \bar{f} \end{cases} \quad (8)$$

where, $P_{c_2} < P_{c_1}$, $P_{m_2} < P_{m_1}$ and $P_{c_1}, P_{c_2}, P_{m_1}, P_{m_2} \in [0,1]$.

The operators expressed above are piecewise linear, but the principles of almost everything's development are nonlinear commonly in the real world. The nonlinearity of the adaptive operators can be comprehended as: in the genetic operating, the protection for the individuals with higher fitness should be strengthened (accelerate the decreasing speed of the probability of crossover and mutation to prevent the missing of the “good” schemas in the evolutionary process), and the increasing speed of the probability of crossover and mutation should be accelerated to make sure that the “bad” schemas can miss more quickly. In order to reach the aim, the authors suggest choose the following operators:

$$P_c = k_1 - k_2 \exp \left\{ k_3 \frac{f' - f_{min}}{f_{max} - f_{min}} \right\} \quad (9)$$

$$P_m = k_1' - k_2' \exp \left\{ k_3' \frac{f_i - f_{min}}{f_{max} - f_{min}} \right\} \quad (10)$$

where , $k_1, k_1', k_2, k_2', k_3, k_3', k_4, k_4', k_c, k_m$ are all constant to be determined.

Similarly, the piecewise form can also be chosen corresponding to the nonlinear situation, as the following operators the authors proposed [6]:

$$P_c = \begin{cases} k_1 - k_2 \exp \left\{ k_3 \frac{f' - \bar{f}}{f_{max} - \bar{f}} \right\}, & f' \geq \bar{f} \\ k_c, & f' < \bar{f} \end{cases} \quad (11)$$

$$P_m = \begin{cases} k_1' - k_2' \exp \left\{ k_3' \frac{f_i - \bar{f}}{f_{max} - \bar{f}} \right\}, & f_i \geq \bar{f} \\ k_m, & f_i < \bar{f} \end{cases} \quad (12)$$

The parameters are also as the former.

4. THE SCHEMA THEOREM OF THE ADAPTIVE GA

4.1 The Linear Operators

Theorem1. (the Schema Theorem of the adaptive GA for the linear operators) Suppose the quality of schema h in population t is $N_h(t)$, after crossover and mutation, the expected quality of individuals which are produced by the i -th individual which belongs to h in the t -th generation and still belong to schema h is $n_i^k(t+1)$, well then the number of individuals belonging to schema h in the t -the generation $N_h(t+1)$ satisfies:

$$N_h(t+1) \geq N_h(t) \frac{\bar{f}_h}{f} \left[1 - \frac{l(h)}{L-1} C_0 - \frac{\delta(h)(f_{\max} - \bar{f}_h)}{(L-1)(f_{\max} - f)} (C_1 - C_0) \right] \times \left[1 - C_0' - (C_1' - C_0') \frac{f_{\max} - \bar{f}_h}{f_{\max} - f} \right]^{\alpha(h)} \quad (13)$$

In order to prove the theorem, two lemmas should be proved first.

Lemma 1: (Chebyshev inequality) For arrays $\{a_i\}, \{b_i\}$, if $a_1 \leq a_2 \leq \dots \leq a_n$ and $b_1 \leq b_2 \leq \dots \leq b_n$ then

$$\frac{1}{n} \sum_{i=1}^n a_i b_i \geq \left(\frac{1}{n} \sum_{i=1}^n a_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n b_i \right). \quad (14)$$

Lemma 2: For m arrays $\{a(1)_i\}, \{a(2)_i\} \wedge \{a(m)_i\}$, if they satisfy at the same time: $a(1)_1 \leq a(1)_2 \leq \dots \leq a(1)_n$, $a(2)_1 \leq a(2)_2 \leq \dots \leq a(2)_n$, $\wedge \wedge \wedge, a(m)_1 \leq a(m)_2 \leq \dots \leq a(m)_n$, then

$$\frac{1}{n} \sum_{i=1}^n a(1)_i a(2)_i \wedge a(m)_i \geq \left(\frac{1}{n} \sum_{i=1}^n a(1)_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n a(2)_i \right) \wedge \left(\frac{1}{n} \sum_{i=1}^n a(m)_i \right) \quad (15)$$

Proof: Based on the Chebyshev inequality,

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n a(1)_i a(2)_i \wedge a(m)_i \\ & \geq \left(\frac{1}{n} \sum_{i=1}^n a(1)_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n (a(2)_i a(3)_i \wedge a(m)_i) \right) \\ & \geq \left(\frac{1}{n} \sum_{i=1}^n a(1)_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n a(2)_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n (a(3)_i a(4)_i \wedge a(m)_i) \right) \\ & \geq \left(\frac{1}{n} \sum_{i=1}^n a(1)_i \right) \cdot \left(\frac{1}{n} \sum_{i=1}^n a(2)_i \right) \wedge \left(\frac{1}{n} \sum_{i=1}^n a(m)_i \right) \end{aligned}$$

The theorem can be proved referring to the approach in literature[3] partly:

Proof:

First, after the selection, the expected value of $n_i^k(t+1)$ is:

$$E(n_i^k(t+1)) = \frac{f_i}{f}. \quad (16)$$

Then consider the effect that the crossover operator imposes on the individuals:

$$n_i^h(t+1) \geq \frac{f_i}{f} \left[1 - \frac{\delta(h)}{L-1} P_c \right] \quad (17)$$

where $L, \delta(h), o(h)$ represent respectively the length of chromosome string, the defined length and order of schema h . And then, consider the mutation operators. For crossover operator and mutation operator affect individuals independently, so:

$$n_i^h(t+1) \geq \frac{f_i}{f} \left[1 - \frac{\delta(h)}{L-1} P_c \right] \times [1 - P_m]^{o(h)} \quad (18)$$

Consulting equation (5) and equation (6), we can learn:

$$\begin{aligned} & n_i^h(t+1) \geq \frac{f_i}{f} \left[1 - \frac{l(h)}{L-1} C_0 - \frac{l(h)(f_{\max} - f_i)}{(L-1)(f_{\max} - f_{\min})} (C_1 - C_0) \right] \times \\ & \left[1 - C_0' - (C_1' - C_0') \frac{f_{\max} - f_i}{f_{\max} - f_{\min}} \right]^{\alpha(h)} \end{aligned} \quad (19)$$

After n_i^h been summarized, inequality can be got from Lemma 2, which is:

$$\begin{aligned} & N_h(t+1) \geq N_h(t) \frac{\bar{f}_h}{f} \left[1 - \frac{l(h)}{L-1} C_0 - \frac{\delta(h)(f_{\max} - \bar{f}_h)}{(L-1)(f_{\max} - f_{\min})} (C_1 - C_0) \right] \times \\ & \left[1 - C_0' - (C_1' - C_0') \frac{f_{\max} - \bar{f}_h}{f_{\max} - f_{\min}} \right]^{\alpha(h)} \end{aligned}$$

In this way, the schema theorem has been proven when the operators are linear.

4.2 The Nonlinear Operators

When the probability operators are nonlinear, linearization is a feasible strategy. As an example let's consider the following adaptive operators:

$$P_c = k_1 \exp \left[-k_2 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right] \quad (20)$$

$$P_m = k_3 \exp \left[-k_4 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right] \quad (21)$$

For P_c :

$$\begin{aligned} P_c &= k_1 \exp \left\{ -k_2 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right\} = k_1 \left\{ 1 - k_2 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} + \left(k_2 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \right)^2 + \dots \right\} \\ &\approx k_1 - k_1 k_2 \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \end{aligned} \quad (22)$$

The similar method can be done for P_m as to transform the original nonlinear operators into the linear ones.

The following operation is similar with that in section 4.1.

It would be much more complex under the situation of piecewise linear adaptive operators in theory because there is a mechanism to sort individuals (The genetic probability of the individuals is a constant when the fitness is lower than the average fitness, but the probability for others is variable dynamically.), so there is a problem of re-sorting in every generation. A method to estimate coarsely is to amplify the probability operators into a constant probability when the

fitness is less than the average fitness as to Holland's schema theorem can be utilized directly.

5 CONCLUSIONS

In the paper, some issues of adaptive GA have been discussed, it contains: three usual forms in adaptive GA were summarized, the GA with adaptive operators were analyzed abstractly, and the schema theorem is presented when the adaptive operators are linear and nonlinear while discussing some problems appearing in some adaptive operators presented in some recent papers.

6 REFERENCES

- [1]. J. H. Holland, *Adaptation in Nature and Artificial Systems*, Arbor: University of Michigan Press, 1975.
- [2]. Xu Zongben, Zhang Jiangshe, Zheng Yalin, *The Bionics in Computational Intelligence*, Beijing: Science Press, 2003.
- [3]. Wang Xiaoping, Cao Liming, *Genetic Algorithm----Theories, Applications and Software realization*, Xi'an: Xi'an Jiaotong University Press, 2002.
- [4]. Chen Changzheng, "Adaptive Methods and Operational Mechanism of Crossover and Mutation Probability in Genetic Algorithm". *Control Theory and Application*, Vol.19, No.1, 2002, pp.41-43.
- [5]. Srinivas M, Patnaik L.M, "Adaptive Probabilities of Crossover and Mutations in Gas", *IEEE Trans. on SMC*, Vol.19, No.1, 2002, pp.41-43.
- [6]. Wang Pan, *Applicational Researches with Soft Computing for some Decision and Control issues*, PhD dissertation, Wuhan: Huazhong University of Science and Technology, 2003.



Zhang Jianjian is a college senior of Wuhan University of Technology. His research interests include computational intelligence etc.



Wang Pan is a Full Associate Professor and a head of Institute of Control and Decision, Wuhan University of Technology. He received the B.S. degree in industrial automation from Wuhan University of Technology, Wuhan, P. R. China, and the M.S. and Ph. D. degrees in systems engineering from Huazhong University of Science and Technology, Wuhan, P. R. China. He has published over 30 Journal papers, 15 Conference papers. His research interests include intelligent control, decision analysis, and biomedical intelligent information systems.

A Class of Accelerated Convergence Algorithms for Solving Ordinary Differential Systems

Dongjin Yuan

Department of Mathematics YangZhou University, Yangzhou, Jiangsu, China

E-Mail: dongjinyuan@yahoo.com

ABSTRACT

In this paper a new technique for acceleration of convergence of discredited waveform method is proposed for solving ordinary linear differential systems. This technique is based on splitting the matrix A of the system in such a way that the resulting iteration matrix has an ideally small spectral radius. Two iterative algorithms are constructed based on the LU and QR decomposition of the system matrix, respectively. Numerical results are reported to compare the convergence properties of this new method with those of the Gauss-Jacobi and the Gauss-Seidel method.

Keywords: ordinary differential equations, waveform relaxation, convergence, decomposition, splitting

1 . INTRODUCTION

In the area of simulation of large electrical circuits, the equation describing the circuit often yields an n -dimensional initial value problem. When simulating a very large scale integrated circuit, the dimension of the problem can be in the millions. One must solve an n -dimensional system of equations at each time point before advancing to the next time point.

In the beginning of the 1980s a new approach for solving these problems was developed at the Electronic Research Laboratory at Berkeley that circumvents these difficulties. In this approach, called the waveform relaxation algorithm, a large amount of computation is required, hence we should consider how to refine the method of waveform relaxation in order to improve the convergence speed of the iteration.

Consider a linear n - dimensional system at the form

$$x'(t) + Ax(t) = f(t), \quad x(0) = x_0, \quad t \in [0, T] \quad (1)$$

where $f \in C([0, T]; R^n)$ and $A \in R^{n \times n}$. The decoupling process can be mathematically described by a splitting of the matrix A as

$$A = M - N, \quad (2)$$

and then the waveform relaxation algorithm is given by

$$\begin{aligned} \frac{d}{dt} x^{(k+1)}(t) + Mx^{(k+1)}(t) &= Nx^{(k)}(t) + f(t), \\ x^{(k+1)}(0) &= x_0, \quad k = 0, 1, \Lambda \end{aligned} \quad (3)$$

The iteration (3), also called the dynamic iteration, is super linearly convergent on any window $[0, T]$ for any splitting (2) (see [1],[7],[9] and [13]). Moreover the algorithms (3) will converge on arbitrarily long windows (see [1]) if

$$\rho((zI + M)^{-1}N) < 1, \quad \text{for all } \operatorname{Re}(z) \geq 0. \quad (4)$$

Unfortunately, this convergence is rather slow for dynamic block Gauss-Jacobi iterations. Thus many techniques were proposed to develop schemes with better convergence

properties. In [3], the techniques of exponential preconditioning and overlapping the components of the system were developed which are quite effective for sparse differential systems resulting from semidiscretization of the heat equation in one or two space dimensions. In [4], the techniques based on rational preconditioning were developed which proved effective for dense differential systems which arise from pseudo spectral discrimination of the heat equation in one space variables. In [5], an approach based on block-Toeplitz preconditioning for both static and dynamic iterations was examined. It was demonstrated that this technique is quite effective in the static case but leads only to modest gains in the rate of convergence in the dynamic case.

For solving linear systems, Yuan [12] has proposed some iterative refinement methods. The convergence properties of these refined methods have better convergence properties than those of the other refinement methods, as the eigenvalues of the iterative matrix can be ideally small. The main purpose of this paper is to establish the corresponding iterative algorithms for solving linear ordinary differential system (1). The main idea is to choose a suitable splitting (2) not only to ensure the iterative procedure (3) satisfies the convergence condition but also to achieve better convergence properties by keeping the spectral radii of the iterative matrix ideally small. In Section 2, we will prove that there do exist at least splitting with $zI + M = DQ$ or $zI + M = QD$, where Q and D are unitary and diagonal, respectively, and with a triangular M , whose spectrum has a given distribution. In Section 3 two algorithms are presented based on theorems in Section 2. Numerical results are reported also in this section.

2 . MAIN RESULT

In this section we will introduce some lemmas and theorems whose proofs are all omitted because of limited space. We denote h the step size, t_0 the starting point of the actual widow, t_r the time point $t_0 + rh$ (we are using equidistant time points), $x_{k,r}$ the k th approximation of the solution of (1) at time t_r , f_r the evaluation of the function f at time t_r and $x(t_r)$ the exact solution of (1) at time t_r .

Lemma 2.1. The following statements are equivalent;

- (a). $x(t)$ is a solution of (1);
- (b). $x(t)$ is a solution of (3);
- (c). $X^* = (x(t)^T, x(t)^T, \dots, x(t)^T)^T$ is the solution of the fixed-point iteration

$$X^{k+1} = HX^k + G\xi + C\eta,$$

with

$$H = \begin{pmatrix} hBN & 0 & \Lambda & 0 \\ hB^2N & hBN & \Lambda & 0 \\ M & M & O & M \\ hB^rN & hB^{r-1}N & \Lambda & hBN \end{pmatrix},$$

$$G = \begin{pmatrix} hB & 0 & \Lambda & 0 \\ hB^2 & hB & \Lambda & 0 \\ M & M & O & M \\ hB^r & hB^{r-1} & \Lambda & hB \end{pmatrix}$$

$$C = \text{diag}(B, B^2, \Lambda, B^r), X^k = (x_{k,1}^T, x_{k,2}^T, \Lambda, x_{k,r}^T)^T,$$

$$\xi = (f_1^T, f_2^T, \Lambda, f_r^T)^T, \eta = (x_0^T, x_0^T, \Lambda, x_0^T)^T,$$

where $B = (I + hM)^{-1}$.

Lemma 2.2. Assume that matrix H has the form in Lemma 2.1 Then

$$\rho(H) = \rho((zI + M)^{-1}N), z = 1/h, R(z) \geq 0.$$

With the above two lemmas, we can prove that there exists an splitting $A = M - N$ of the matrix A in the system (1) such that the spectral radius of the matrix $((zI + M)^{-1}N)$ can be any given parameters.

Lemma 2.3 Suppose that $(zI + A)$, with $R(z) \geq 0$, and Q are $n \times n$ (complex) nonsingular matrices, and $\lambda_1, \lambda_2, \Lambda, \lambda_n$ are complex numbers such that

$$\lambda_j \neq 1, j = 1, 2, \Lambda, n.$$

Then there exist matrices M and N such that

- (a). $A = M - N$;
- (b). $(zI + M)$ is nonsingular;
- (c). the eigenvalues of $((zI + M)^{-1}N)$ are $\lambda_1, \lambda_2, \Lambda, \lambda_n$;
- (d). the columns of Q are eigenvectors of $((zI + M)^{-1}N)$ corresponding to $\lambda_1, \lambda_2, \Lambda, \lambda_n$, respectively.

The following two Lemmas (see [14]) will show that $((zI + M)$ in Lemma 2.3 can be triangular or scaling unitary.

Lemma 2.4. For every nonsingular upper triangular $n \times n$ matrix V and given numbers $\lambda_1, \lambda_2, \Lambda, \lambda_n$, there exist $n \times n$ invertible diagonal matrix D and matrix T , such that

$$DV = T^{-1}(I - \Lambda)T. \quad (5)$$

where $\Lambda = \text{diag}(\lambda_1, \lambda_2, \Lambda, \lambda_n)$.

Lemma 2.5. Let U and L be matrices upper and lower triangular, respectively, where U is nonsingular, and let given numbers $\lambda_1, \lambda_2, \Lambda, \lambda_n$. Then there exist nonsingular matrices T and D , where D is diagonal and with positive elements, such that

$$U^{-1}DL^T = T(I - \Lambda)T^{-1}. \quad (6)$$

From now on we will give three main theorem which are the base of new algorithms to improve the convergence properties of waveform relaxation methods.

Theorem 2.1. Let $(zI + A)$ be an $n \times n$ nonsingular matrix having LU decomposition. Then there exists a triangular splitting of A , given by

$$A = M - N$$

Such that the spectral radius $\rho((zI + M)^{-1}N)$ can be arbitrary small. In particular, for $\Lambda = \text{diag}(\lambda_1, \lambda_2, \Lambda, \lambda_n)$ the matrix M can be chosen such that the spectrum of

$(zI + M)^{-1}N$ is the same as that of Λ .

For the convergence of splitting methods, we have to choose all $|\lambda_i| < 1$. Then we have the following corollary.

Corollary 2.1. Let $(zI + A)$ be an $n \times n$ matrix having an LU decomposition. Then there exist a convergent splitting of A with type (2) in which M is lower triangular. Moreover, for any real number $\rho \in (0, 1)$ there exists such a splitting for which the spectral radius $\rho((zI + M)^{-1}N) = \rho$.

Next we will show that the splitting matrix M can be symmetric and $(zI + M)$ be a positive definite matrix. In this case, the spectral radius of $(zI + M)$ can again be arbitrary small.

Theorem 2.2 Let $(zI + A)$ be $n \times n$ nonsingular matrix having LU decomposition and $\lambda_1, \lambda_2, \Lambda, \lambda_n$ be real numbers. Then there exists a splitting of A of type (2) such that $(zI + M)$ is symmetric and positive definite. Moreover, for any $\rho \in (0, 1)$, M can be chosen such that

$$\rho((zI + M)^{-1}N) = \rho.$$

In theorem 2.1 and 2.2 we have discussed the decomposition of LU type. The following theorem shows that this theory is also true for the decomposition of QR type.

Theorem 2.3 For every $n \times n$ nonsingular matrix $(zI + A)$, and every choice of numbers $\lambda_1, \lambda_2, \Lambda, \lambda_n$ satisfying $\lambda_j \neq 1, j = 1, 2, \Lambda, n$, there exists a convergent splitting of A of type (2) such that $(zI + M)$ is scaling unitary and $\lambda_i, i = 1, 2, \Lambda, n$ are the eigenvalues of $((zI + M)^{-1}N)$.

3. ALGORITHMS AND NUMERICAL EXAMPLES

It follows from Corollary 2.1 and Theorem 2.3 that we can construct new approaches to iterative refinement of linear ordinary differential systems. In this section we will not consider pivoting (or permutation) for an LU decomposition of matrix $(zI + A)$. However, all results in this section hold for LU decomposition with pivoting as well.

In order to guarantee the convergence, we take $|\lambda_i| < 1$, for example,

$$\lambda_i = 1/(i + 10), i = 1, 2, \Lambda, n,$$

Then $\rho((zI + M)^{-1}N) < 0.1$. Hence we can choose

$$d_i = (1 - \lambda_i)/u_{ii}, i = 1, 2, \Lambda, n.$$

From Corollary 2.1, we have that

$$(zI + M)^{-1}N = I - DU.$$

Now we give an iterative refinement algorithm for waveform relaxation method with a single step as follows

Algorithm 3.1

1. For matrix A and chosen z , decompose $(zI+A)$ into LU , i.e., $(zI+A) = LU$.
2. For chosen λ_i and the diagonal elements u_{ii} of $U, i = 1, 2, \Lambda, n$, compute

$$\lambda_i = (1 - \lambda_i) / u_{ii}, i = 1, 2, \Lambda, n.$$

3. Set

$$M_{LU} = LD^{-1} - zI, N_{LU} M_{LU} - A,$$

$$\text{Where } D = \text{diag}\{d_1, d_2, \Lambda, d_n\}.$$

4. For an initial value $x(0) = x_0$, iterate

$$\begin{aligned} \frac{d}{dt} x^{(k+1)}(t) + M_{LU} x^{(k+1)}(t) = \\ N_{LU} x^{(k)}(t) + f(t), k = 0, 1, \Lambda \end{aligned} \quad (7)$$

End of algorithm 3.1

Similarly, for the QR factorization where R is triangular, there is another iterative refinement for waveform relaxation method.

Here we also take λ_i such that $|\lambda_i| < 1, i = 1, 2, \Lambda, n$ to guarantee the convergence of the method. For simplicity we can decide

$$d_i = (1 - \lambda_i) / r_{ii}, i = 1, 2, \Lambda, n.$$

Then the corresponding algorithm is the following

Algorithm 3.2

1. For given matrix A and chosen z , decompose $(zI+A)$ into QR , i.e., $(zI+A) = QR$,

Where Q is unitary and R is triangular.

2. For chosen λ_i and the diagonal elements r_{ii} of $R, i = 1, 2, \Lambda, n$, compute

$$d_i = (1 - \lambda_i) / r_{ii}, i = 1, 2, \Lambda, n.$$

3. Set

$$M_{QR} = QD^{-1} - zI, N_{QR} = M_{QR} - A,$$

$$\text{Where } D = \text{diag}\{d_1, d_2, \Lambda, d_n\}.$$

4. For an initial value $x(0) = x_0$, iterate

$$\frac{d}{dt} x^{(k+1)}(t) + M_{QR} x^{(k+1)}(t) = N_{QR} x^{(k)}(t) + f(t), k = 0, 1, \Lambda \quad (8)$$

End of algorithm 3.2

As an example, consider the linear ordinary differential system (1) with system matrix

$$A = 100 \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}.$$

If the implicit Euler method is used with step size $h = 0.01$, then $z = 1/h = 100$ and

$$(zI + A) = \begin{pmatrix} 300 & -100 & 0 & 0 \\ -100 & 300 & -100 & 0 \\ 0 & -100 & 300 & -100 \\ 0 & 0 & -100 & 300 \end{pmatrix}.$$

Decompose $(zI + A) = LU$, where

$$L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1/3 & 1 & 0 & 0 \\ 0 & -3/8 & 1 & 0 \\ 0 & 0 & -8/11 & 1 \end{pmatrix},$$

$$U = \begin{pmatrix} 300 & -100 & 0 & 0 \\ 0 & 800/3 & -100 & 0 \\ 0 & 0 & 2100/8 & -100 \\ 0 & 0 & 0 & 5500/21 \end{pmatrix}.$$

If applying Algorithm 3.1 with λ_i , chosen as

$$\lambda_1 = 1/2001, \lambda_2 = 1/3001, \lambda_3 = 1/4001, \lambda_4 = 1/5001,$$

The

$$M = \begin{pmatrix} 4003/20 & 0 & 0 & 0 \\ -2001/20 & 15008/90 & 0 & 0 \\ 0 & -3001/30 & 52021/320 & 0 \\ 0 & 0 & -4001/40 & 34011/210 \end{pmatrix}$$

$$N = \begin{pmatrix} 3/20 & 100 & 0 & 0 \\ -1/20 & -2992/90 & 100 & 0 \\ 0 & -1/30 & -11979/320 & 100 \\ 0 & 0 & -1/40 & -7989/210 \end{pmatrix}.$$

For this given test equation, if the Gauss-Seidel method is used, then

$$\rho((zI + M)^{-1}N) = \frac{3 + \sqrt{5}}{18} \approx 0.2909.$$

Thus the convergence speed of Algorithm 3.1 is faster than that of the Gauss-Seidel method in this case.

Now numerical results are reported in this section to compare the convergence properties. The test equation is derived from the heat equation. Discretizing the heat equation in one space variable, given by

$$\frac{\partial y}{\partial t} = a^2 \frac{\partial^2 y}{\partial x^2}, \quad (10)$$

Leads to the linear system of the form

$$y'(t) + Qy(t) = 0, t > 0, y(0) = y_0, \quad (11)$$

$$y = [y_1, y_2, \Lambda, y_n]^T = [y(t, x_1),$$

$$y(t, x_2), \Lambda, y(t, x_n)]^T, n = 100,$$

$$Q_2 = \frac{a^2}{h_x^2} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & 0 & 0 & 0 \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix} \in R^{n \times n},$$

and h_x is the step size for discretizing the spatial variable x . Here $a=1$.

The differential system (11) is solved by the Gauss-Jacobi, Gauss-Seidel, and the algorithm 3.1 with $n=100$, the dimension of the system, and with different length of window. Here the implicit Euler method is used with the initial solution $y^{(0)}(t) = y_0$. The parameters λ_i in algorithm 3.1 is given by

$$\lambda_i = \frac{1}{i + 199999}.$$

With the step size $h_y = 0.1$ and $h_t = 0.01$, the spectral radii of the iterative matrices are given by

Gauss-Jacobi method: $\rho((I/h + M)^{-1}N) = 0.6663$;

Gauss-Seidel method: $\rho((I/h + M)^{-1}N) = 0.4440$;

Refined method: $\rho((I/h + M)^{-1}N) = 0.1090$.

Table 1 gives the number of iterations needed to reduce the errors below the required tolerances, $TOL=10^{-4}$ and 10^{-8} on the window $[0,0.25]$, $[0,0.5]$, $[0,1]$ and $[0,2]$. For this test equation, the convergence rate of the algorithm 3.1 is faster than that of Gauss-Seidel iteration.

Table 1: Numbers of iterations with different window size T

	T=0.25		T=0.5		T=1		T=2	
	10^{-4}	10^{-8}	10^{-4}	10^{-8}	10^{-4}	10^{-8}	10^{-4}	10^{-8}
	79	123	135	194	242	322	447	561
Gauss-Jacobi								
Gauss-Seidel	46	71	75	109	131	177	237	301
Algorithm 3.1	33	51	52	75	88	119	156	199

Remark 3.1 We should point out that the condition number of the iterative matrix of algorithm 3.1 will be large as the dimension of the system increases. Thus the calculation of the spectral radius will be influenced by the round-off errors. It is possible that the practical spectral radius of Algorithm 3.1 is larger than the theoretical value.

4 REFERENCES

- [1] K. Burrage, Parallel and Sequential methods for Ordinary Differential Equations, Oxford University press, Oxford, 1995.
- [2] K. Burrage, C. Dyke and B. Pohl, On the performance of parallel waveform relaxations for differential systems, Applied Number Maths, 20(1996), 39-55.
- [3] K. Burrage, Z. Jackiewicz, S.P.Norsett and R. A. Renaut, Preconditioning waveform relaxation Iteration for differential systems, BIT, 36(1996), 54-76.
- [4] K. Burrage, Z.Jackiewicz and R.A.Renaut, Waveform relaxation techniques for pseudospectral methods, Numerical methods for Partial Differential Equations, 12(1996), 245-263.
- [5] K. Burrage, Z.Jackiewicz and B.Welfert, Block-Toeplitz preconditioning for static and dynamic linear systems, Linear Algebra Appl., 279(1998), 51-74.
- [6] Z. Jackiewicz and M.Kwapisz, Convergence of waveform relaxation methods for differential-algebraic equations, SIAM J. Numer. Anal., 33(1996), 2303-2317.
- [7] R.Jetsch and B. Pohl, Waveform relaxation with overlapping splittings, SIAM J. Sci. Comput., 16(1995), 40-49.
- [8] P. J. Lanzkron, D.J. Rose and D.B.Szyld, Convergence of nested classical iterative methods for linear systems, Numer. Maths., 58(1991), 685-702.
- [9] B. Leimkuhler, Estimating waveform relaxation convergence, SIAM J. Sci. Comput., 14(1993), 872-889.
- [10] B. Pohl, On the convergence of the discretized multisplitting waveform relaxation algorithm Applied Number. Math., 11(1993), 251-258.
- [11] D. Yuan and K. Burrage, Convergence of the parallel chaotic waveform relaxation method for stiff systems, J. Comput. Appl. Math. 151(2003), 201-213
- [12] J. Yuan, Iterative refinement using splitting methods, Linear, Algebra Appl., 273(1998),199-214.
- [13] Z.Zlatev, Computational methods for general sparse matrices, Kluwer academic, 1991.

Genetic Algorithm and Evolutionary Programming: A comparison study

Wei Gao

Wuhan Polytechnic University, Wuhan, Hubei 430023, P. R. China

Email: wgaowh@hotmail.com Tel.: +86-27-83950207

ABSTRACT

Genetic algorithm and evolutionary programming are two generally used evolutionary algorithms. Due to the difference of their origin, there are a lot of differences between their biologic bases, algorithm operation and some other operational details. So, the performances of the two algorithms are different. In this paper, these differences are analyzed comprehensively by theory and revealed by simulation experiments. The results show that the performance of evolutionary programming is better than that of genetic algorithm and the evolutionary programming is more suitable for practical applications.

Keywords: Genetic algorithm, Evolutionary programming, Comparison.

1. INTRODUCTION

As a good global optimal tool, evolutionary algorithm (EA) or evolutionary computation (EC) is widely used in management, mathematics, electronics, computer science, civil engineering and other engineering fields [1]-[2]. So, the study and application of the evolutionary algorithm are very hot spot in social science, natural science and engineering science. As the two typical algorithms, genetic algorithm and evolutionary programming are widely used. As a global optimal tool, they can substitute each other. But due to the difference of their

origin, their performances are different. In order to apply the two algorithms suitably, from the biologic bases, algorithm operation, et al, the two algorithms are compared comprehensively in this paper.

This paper is organized as follows. In section 2, we introduce the two algorithms briefly. Section 3 describes the comparison study of the two algorithms. In section 4, the simulation experiments are described. The conclusions are presented in section 5.

2. GENETIC ALGORITHM AND EVOLUTIONARY PROGRAMMING

Genetic Algorithm (GA)

Genetic algorithm is proposed by American scholar J. H. Holland in 1960's [3]-[5]. But for its study history, the study of genetic algorithm should start from the simulation of biological evolution by computer in 1950's. But at that time, this study is restricted in biological field. At the beginning of 1960's, Holland discovered the similarity between adaptability of system to environment and biological evolution when he studied the self-adaptive system. From this similarity, Holland proposed a computational method, which simulates the biologic evolution, and called it genetic algorithm. From that time, genetic algorithm has developed very quickly and become the most widely used evolutionary algorithm. The basic flow chart of genetic algorithm is showed in Fig.1.

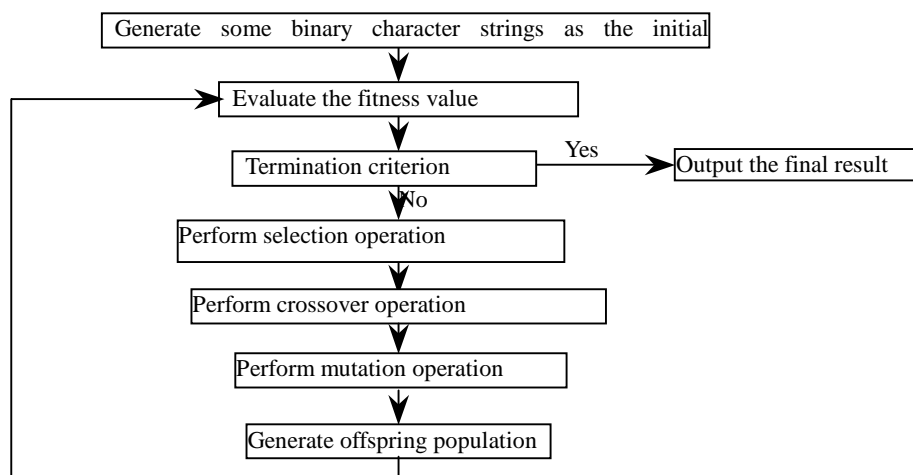


Fig. 1 Basic flow chart of genetic algorithm

As a branch of evolutionary algorithm that is studied most thoroughly, the biologic base of genetic algorithm is stable. The basic idea of genetic algorithm comes from Darwinism and Mendel's genetics. In Darwinism, the process of biologic evolution is an adaptive process to environment, which shows "superior win and inferior eliminate". The basic characteristics

of every individual in the species are inherited by its offspring. But due to casual factors, the offspring is not the same as its parent. If these varieties of the offspring can fit with the environment, they can be reserved. In one environment, only those individual characteristics that fit with the environment more suitable can be reserved, which is called "winner survive"

principle or natural selection process. In Mendel's genetics, inheritance instruction code is sealed in every cell and included in chromosome as gene. Each gene has its special position and controls one special characteristic. The offspring produced by one gene has some adaptability to the environment. The crossover and mutation of the gene may produce the offspring that fit the environment more suitable. Through the natural selection, the gene structure, which fit the environment more suitable, can be reserved.

Evolutionary Programming (EP)

Evolutionary programming is based on Finite State Machine (FSM) model in its early stage, which is proposed by L. J. Fogel in 1960's when he studied the artificial intelligence. In this early model, the state of the machine is mutated by evenly distributing rules [6]-[7]. In 1990's, the evolutionary programming was developed by D. B. Fogel and was made to solve the optimal problems in real space. In this improved

model, the mutation operator is based on normal school. So, the evolutionary programming has been become an optimal tool and was used in many practical problems [7]-[8]. Different from genetic algorithm, evolutionary programming simulates the evolution of species not the genes; so evolutionary programming emphasizes the linking of species in its evolutionary process. It is said that, the evolution of genetic algorithm only constructs the inherit linking between parent and offspring, or good parent generate good offspring certainly. While evolutionary programming emphasizes the behavior evolution of the species, and it constructs the behavior linking between parent and offspring, or good offspring can survive and not considering its parent. So, the fitness value of genetic algorithm is to select the parent while that of evolutionary programming is to select offspring. The basic flow chart of evolutionary programming is showed in Fig.2.

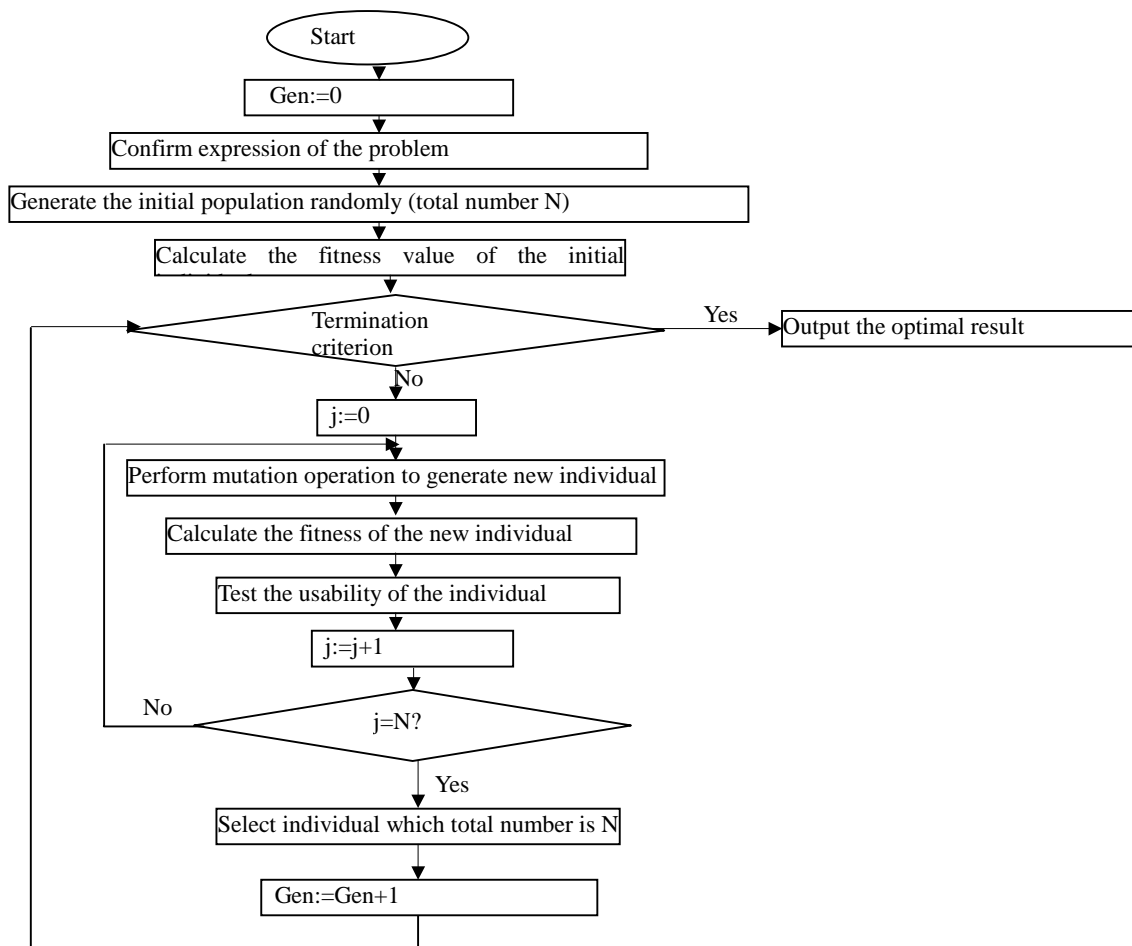


Fig. 2 Basic flow chart of evolutionary programming

Different from the basic algorithm, in our evolutionary programming, author adds the step of testing the usability of individual. Because the mutation in evolutionary programming is a random disturbing to the individual, it cannot guarantee the individual mutated is in its original range. This problem is often ignored in other literatures.

3. COMPARISON STUDY

Genetic algorithm and evolutionary programming are two main methods in evolutionary algorithm. Their ideas all come from natural biologic evolution. They perform optimal operation by population method and are all global optimal tools. From the idea of simulating evolutionary optimization, there are a lot of similarities between them, and they can substitute each other in some optimal problems. But because the two algorithms simulate the different aspects of the biologic evolution, there are some different methods in their

operation. Because there are large difference in their biologic bases and algorithm operation, they show different efficiency and effect when they are used in practical problems. In order to apply them suitably in real problems, the comparison study of the two algorithms is done as follows.

Biologic Bases

The basic idea of genetic algorithm comes from Darwinian evolution and Mendelian genetic theories.

From the study of cytology and genic theory, we know that there are two microcosmic mechanisms (inherit and mutation) in biologic evolution, and they are all occurred in chromosome of biologic cell. The chromosome is mainly composed of DNA and protein. DNA is a kind of macromolecule compound and it is the main hereditary substance. The substance unit that controls the biologic heredity is called gene, which is the section of DNA. The genes are arranged as a line on the chromosome. The sequence of the genes arrangement represents the hereditary information. Through the copy and crossover of the genes that are, gene separation, gene combination and gene concatenate interconversion, the heredity of biologic characters is selected and controlled. At the same time, through gene recombination, gene mutation and mutation of structure and number of chromosome, the rich and colorful biologic world is generated. The recombination of biologic genes is the microcosmic mechanism of the biologic reproduction essentially. The genes recombination that comes from mating of sexual reproduction is the main method of biologic evolution.

In the natural selection theory of Darwinian evolution, to survive, a living creature must compete with other living creatures. The survival competition includes three types, which are competition in species, competition between species and competition with inorganic environment. In survival competition, the individual that possesses of the favorable characters may be survived in larger probability and inherit its characters to the offspring. By contrary the individual that possesses of the unfavorable characters may be eliminated in larger probability and produce offspring in small probability. So, the winner individual in survival competition is the more favorable individual to environment. Darwin calls this process of survival of the fittest as natural selection. The living creatures in the natural world evolve as the process of this natural selection.

According to above biological theory, genetic algorithm performs its algorithm operation of coding, copy, crossover and mutation.

Different from genetic algorithm, the biologic base of evolutionary programming is non-Darwinian evolution. According to the theory of modern molecular biology, it is a kind of neutral theory of biological evolution [9]. In this theory, the adaptive evolution based on Darwinian natural selection is a nonessential evolution of the living creatures. The precondition of the biologic evolution is no selection and non-selection can also produce biologic evolution. Moreover extremely adaptive selection is an impasse, which is the biological reason of premature convergence of genetic algorithm. As to the molecular biology, the evolution of the living creatures is a haphazard process that isn't been disturbed by selection. The success of each evolutionary progress is the base for the next evolutionary progress. The implement of evolution is the accumulating result of those random evolutionary progresses. The function of natural

selection is only a kind of modifying to this progress, and makes the detracted evolutionary states convergence. This thinking of biologic evolution is similar to the implement process of evolutionary programming [10].

Furthermore, the Lamarckism is also combined in evolutionary programming, that is to say, if the changed character to survive of the living creature is more compatible with the environment, it can be inherited to the next generation. According to above biological theory, evolutionary programming performs its algorithm operation.

Algorithm Operation

Due to the difference of two algorithms' biologic bases, there exist larger differences in their algorithm operation. First, genetic algorithm's operation must be done under one coding mode, or the object of optimal problem must be coded and changed to code string, that is to say, the operation of genetic algorithm is done in manner of the chromosome. While, the optimal object of evolutionary programming needs not to be coded. So, the optimal object in genetic algorithm is genotype, while that in evolutionary programming is representational type. The above feature makes the operation of evolutionary programming powerful. The user cannot be confined to one kind of coding mode, which makes the application of evolutionary programming more flexible. Second, because of the complicated biological base of genetic algorithm, its operation includes three steps, which are crossover, mutation and selection. While there are only two steps, which are mutation and selection, in evolutionary programming' operation, and there isn't crossover operator that mimics the mating of sexual reproduction. In the genetic algorithm, crossover is the main evolutionary operation, and mutation is an operation, which only maintains the diversity of population. By contrary, in evolutionary programming, mutation is the only evolutionary operation, which is required to research not only globally but also locally. Moreover, there requires two parameters to control the operation, which are crossover probability and mutation probability, in genetic algorithm. The performance of the algorithm is affected mostly by two parameters. But recently, there isn't an appropriate method to choose them. Generally, the two parameters are chosen by user's experience, so it's hard to avoid subjectivity in the operation of genetic algorithm. By contrary, there are not those parameters in operation of evolutionary programming, which makes its application more flexible. Third, the operators of two algorithms are different. The selection operator of genetic algorithm is a random selection method, and the selection probability is usually proportion to the individual's fitness value. Also, the offspring population is confined to be selected from its parent population. While in evolutionary programming, the selection operator is a kind of stochastic tournament modal that is called random competition with parameter q . In this operation, the next population is selected from the combined set of parent population and its offspring population. The mutation operator of genetic algorithm is usually a kind of reversion of symbol, while that of evolutionary programming is a kind of variance diversification. At last, because the operation of genetic algorithm is done in genotype, then it emphasizes the chromosome linking of individuals', that is to say, the evolution of genetic algorithm is based mainly on chromosome chain. On contrary, the operation of evolutionary programming is the process of the evolution of population, and it emphasizes the behavior linking of the individuals' or its operation is based mainly on behavior chain.

Algorithm Course

First, genetic algorithm is a process of from bottom to top, and it is the process of combination of building blocks of parameter codes. While, evolutionary programming is a process of from top to bottom. Second, it is very complicated to cope with constrain conditions in genetic algorithm. Generally, the punish function method is applied in practical problems, but the design of punish function is a problem that is very hard to be solved. In evolutionary programming, it is easy to cope constrain conditions. It is a process to test the feasibility of individual's fitness value. Because the mutation of evolutionary programming is essentially a random disturbance to original solution, it cannot guarantee the mutated new solution in search range, that is to say, the operation of evolutionary programming may produce the infeasible solution. The existence of infeasible solution can not only make the final result wrong but also make the efficiency of algorithm very low. So, the process of testing the feasibility of new solution is very necessary. At the same time, constrain conditions is a restriction to the feasibility of new solution, it can be put into the process of testing new solution very easily. Third, there is a strong trend to convergent to local range in genetic algorithm. It makes the phenomenon of premature convergence very severely and hard to overcome, especially to problem with large search range. Contrarily, there isn't this phenomenon in evolutionary programming.

So, by comparison of the two algorithms, the evolutionary programming shows its obvious superiority. As D. B. Fogel has said [8], evolutionary programming mimics the natural evolution more rationally, because in natural evolution, it is very important to maintain the behavior linking of parent population and offspring population, and in natural evolution it is the evolution of individual's behavior but not its coding. Natural selection is to select the behavior mutation, and the evolution of chromosome structure is only the result of behavior selection.

4. SIMULATION EXPERIMENT

In order to compare the optimal capability of genetic algorithm and evolutionary programming, in this comparison study, the Simple Genetic Algorithm (SGA) and standard Evolutionary Programming are used.

In simulation experiment, the numerical function is Schaffer function. Its expression is as follows:

$$f(x_1, x_2) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1.0 + 0.001(x_1^2 + x_2^2)]^2}$$

The search range is $-100 \leq x_1, x_2 \leq 100$.

This function has infinite extremum, but only one (0, 0) is global maximum point, where the function value is 1. The character of the function is that there exists a circular raphe around the global maximum point, where the function value is 0.990284. So, when optimizing this function, the optimal method is often trapped to local maximum. The generally used optimal method is often invalidated to this function. Therefore, this function has become a standard problem to test the performance of evolutionary algorithm, and has been used in many studies. The distributing map of this function is showed

in Fig. 3.

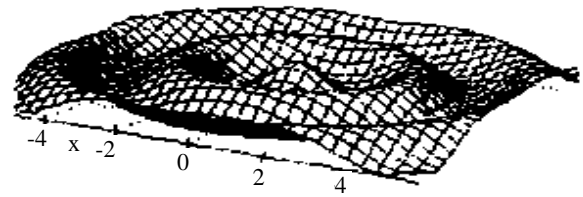


Fig. 3 The distributing map of testing function

Another character of this function is that the range of parameters is very large. As the previous study showing, the range of parameters has strongly affected to the performance of evolutionary algorithm. So, the large range of parameters of Schaffer function is a large difficulty to many evolutionary algorithms. In order to illuminate the problem and make the computing simple, here the results of ten independent computing of the two algorithms is used in comparison study and the threshold value of evolutionary algorithm is 0.999, that is to say, when the result of evolutionary algorithm reaches this value, we think this algorithm convergent.

When using simple genetic algorithm, the range of parameters is $[-100, 100]$, and the population size is 100. At the condition of threshold value of evolutionary generation is 1000, the computation of ten times are all premature convergence. The optimal result is unsuccessful. When the population size is 1000 and the other conditions are as same as above, in ten independent computation times, only one time is convergent to optimal threshold value. The convergent result is that the evolutionary generation is 1730, the optimal result is 0.99995, and the other computing results are all premature convergence.

When using standard evolutionary programming, at the same conditions, when population size is 100, three times of computation in ten times reach the optimal threshold value. The average result of three computing times is that the evolutionary generation is 467 and the optimal result is 0.9997. The other experiments are premature convergence or do not reach the optimal threshold value when evolutionary generation is 1000. When population size extends to 1000, ten computing experiments all reach the optimal threshold when evolutionary generation is 1000. The best result is that the optimal value is 0.99997 when evolutionary generation is 50. The average evolutionary convergent generation is 423.

From the above computing results, we can conclude that evolutionary programming can avoid the premature convergence very well and its performance is better than genetic algorithm's.

In order to study the influence of range of optimal parameters on evolutionary effect, the below experiments are carried out. The two parameters' ranges of above function are all compressed to $[-10, 10]$, and the other experiment conditions are as same as above. When applying simple genetic algorithm, as the population size is 100, ten times of computing experiments are all premature convergence. While as population size is expanded to 200, only one time of experiment in ten times reach the optimal threshold value, and its result is that evolutionary generation is 1220, and optimal result is 0.99992. While using evolutionary programming, as

population size is 100, six times of computing experiments in ten times reach optimal threshold value. The average result is that evolutionary generation is 310 and optimal result is 0.9995. When population size is compressed to 50, five times of computing experiments in ten experiments reach the threshold value of 0.999. Their average results are that evolutionary generation is 504 and optimal result is 0.9997.

In order to compare the optimal effect of two algorithms under small parameter range, the parameters' ranges are compressed to $[-2,2]$. When using genetic algorithm, as population size is 100, three times of experiments are premature convergence and seven times reach optimal threshold. Their average result is that evolutionary generation is 485 and optimal result is 0.9997. As population size is 50, three times of experiments reach optimal threshold. The average result is that evolutionary generation is 553 and optimal result is 0.9997. When using evolutionary programming, as population size is 50, all ten times of experiments are convergent to optimal threshold. The average result is that evolutionary generation is 124.7 and optimal result is 0.9996.

From the above computing results we know that the range of optimal parameters has large influence on optimal effect of evolutionary algorithm. When parameter range is large, the optimal outcome is bad. While when parameter range is small, the performance of algorithm improves very greatly. Comparing the results of genetic algorithm and evolutionary programming, we can see that evolutionary programming can avoid premature convergence remarkably and improve the stability of algorithm. Moreover, the convergent speed of evolutionary programming is more quickly than genetic algorithm's. As the above experiment shows, at the condition that parameters' range is $[-2,2]$ and population size is 50, the average convergent generation of evolutionary programming is 124.7, while genetic algorithm is 353 and is 2.83 times of evolutionary programming.

From the above numerical experiments, we can draw the follow conclusions.

- a. The range of optimal parameters has large influence on optimal effect of evolutionary algorithm. The parameter range is larger and the optimal effect is poorer. So, in practical application, we should compress the optimal range as soon as possible to improve the optimal effect of evolutionary algorithm.
- b. The optimal performance of evolutionary programming is better than genetic algorithm. Its optimal result is more stable and evolutionary programming can avoid premature convergence very affectively.
- c. At the same conditions, the optimal speed of evolutionary programming is more rapidly, that is to say, the threshold optimal generation is smaller.

So, the results of numerical experiment verified the theoretic analysis of section 3.

5. CONCLUSION

Genetic algorithm and evolutionary programming are two typical methods of evolutionary algorithms. Due to their good optimal capability, they have been used in practical problems

largely. Because of the difference of their origin and principles, there exist much difference in their algorithm design, algorithm operation and optimal results. In this paper, from biological bases, algorithm operation and algorithm course, we compare two algorithms theoretically and comprehensively. At last, through a numerical experiment, the operation performance of two algorithms is analyzed comprehensively. From the whole analysis, we can conclude that the operation of evolutionary programming is easier and its application more flexible. Also, the computing speed and stability of evolutionary programming are all superior to genetic algorithm. So, evolutionary programming is a more perfect algorithm in practical applications.

6. REFERENCES

- [1] D. Dasgupta, Z. Michalewicz, Evolutionary Algorithm-An Overview, In: D. Dasgupta, Z. Michalewicz Eds. *Evolutionary Algorithms in Engineering Applications*. Berlin: Springer, 1997, pp. 3~28.
- [2] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, New York: Spinger-Verlag, 1996.
- [3] M. Srinivas, L. M. Patnaik, "Genetic Algorithms: A Survey", *IEEE Computer*, Vol.27, No.6, 1994, pp. 17~26.
- [4] A. J. Keane, "Genetic Algorithm Optimization of Multi-peak Problems: Studies in Convergence and Robustness", *Artificial Intelligence in Engineering* Vol.9, 1995, pp. 75~83, 1995.
- [5] A. M. Zalzal, P. J. Fleming, *Genetic Algorithms in Engineering Systems*, London: The Institution of Electrical engineers, 1997.
- [6] D. B. Fogel, K. Chellapilla, Revisiting Evolutionary Programming, In: S. K. Rogers, D. B. Fogel, J. C. Bezdek, et al Eds, *Proc. of Applications and Science of Computational Intelligence*. Orlando, Florida, 1998, pp. 2~11.
- [7] D. B. Fogel, L. J. Fogel, An Introduction to Evolutionary Programming, In: J. M. Alliot, E. Lutton, et al Eds, *European Conf. on AE'95*, Berlin: Springer, 1996, pp. 21~33.
- [8] D. B. Fogel, "Applying Evolutionary Programming to Selected TSP", *Cybernetics and Systems*. Vol.24, 1993, pp. 27~36.
- [9] M. Kimura, *The Neutral Theory of Molecular Evolution*, Cambridge: Cambridge University Press, 1983.
- [10] G. B. Fogel, D. B. Fogel, "Continuous Evolutionary Programming: Analysis and Experiments", *Cybernetics and Systems*, Vol.26, 1995, pp.79~90.



Wei Gao is an Associate Professor and a Doctor in Wuhan Polytechnic University. He graduated from China University of Mining and Technology in 1995. He has published over 40 Journal papers and conference papers. His research interests are in distributed parallel computation, intelligent computation, etc.

A High-efficient Parallel Reasoning Algorithm

Minghe Huang, Cuixiang Zhong
Software College, Jiangxi Normal University
Nanchang, Jiangxi Province, 330027, China

Email: Huangmh0093@sina.com Tel: 8507920(O), 8506927(H)

ABSTRACT

The speed and efficiency of parallel reasoning has become one of the hot problems in parallel processing domain. Although some people have already propounded some parallel reasoning algorithms on some parallel models, these algorithms still have some shortcomings such as their reasoning search is blind and load balancing is not proper. In order to overcome these shortcomings, the paper designs a parallel reasoning algorithm with heuristic reasoning search and proper load balancing, it has really improved the efficiency of parallel reasoning.

Keywords: knowledge-based reasoning, parallel reasoning, reasoning search, load balancing, MIMD shared-memory model, parallel reasoning algorithm.

1. INTRODUCTION

Knowledge-based reasoning is a key technology in artificial intelligence, expert systems and logical programming. With the fast development of parallel processing, how to improve the speed and efficiency of parallel reasoning on a parallel system with multiple processors has become a hot problem in the domain of parallel processing. The research of the parallelization of rule-based reasoning as a common reasoning method is especially significant.

Production-rule-based reasoning is divided into two classes: forward reasoning and backward reasoning. Among them, forward reasoning is essentially a process of extending and searching a state-space-tree. The root of the tree corresponds to the initial set of facts or the set of initial state (the problem to be solved). The inner nodes of the tree corresponds to the sets of intermediate states which produce intermediate conclusions. The leaf nodes are divided into two classes: (1) the nodes containing not goal state; (2) the nodes containing goal states, which means the search has found a solution and is ok if we need only one solution or otherwise it should be continued until all queue-stacks are empty. The reasoning process, according to the facts represented by the current state node, selecting the remaining rules that match the current facts and haven't been used previously from the rulebase, is to apply these rules respectively to the current state node in order to obtain different new state nodes as the expansion nodes of the current state node.

If the problem is very complex, a sequential reasoning search is usually time consuming. So in order to speed up reasoning and improve its efficiency, it is significant to parallelize the reasoning. In 1995, professor Chen Huaping and etc proposed a parallelized reasoning algorithm on a MIMD shared-memory model. The reasoning process realized in the algorithm consists of two phases: The search phase in which each processor performs a state space search on separate parts of a search tree; the second phase is the

dynamic load balancing phase, which is performed only when a processor's queue-stack is empty and also used to maintain maximal processor utilization during the search phase. The search and load balancing phases are performed alternately until some processor finds a goal node during the search phase. But there are two shortcomings in this algorithm: one is the reasoning search is more blind, the nodes with higher certainty fact haven't been given priority in search, which makes the reasoning slow; another is the load-balancing algorithm hasn't been designed properly, it requires the processor with heavier load to stop to transfer some nodes from its queue-stack to other processors with lighter load, which makes the speed of reasoning much lower and that violates a taboo of parallel processing.

Hence we have improved their algorithms and designed a parallel algorithm with heuristic search and proper load balancing algorithm, that is in the improved algorithm, we have added a step of sorting the generated nodes according to their certainty facts, which puts the nodes with higher certainty fact on the top of the queue-stack, and thus realizes the heuristic reasoning search. In addition, the algorithm makes it a rule that only if a processor's queue-stack is empty can it move some nodes from the processors with heavier loads, which thus have realized ingeniously the parallelization of multiple processors and improved the efficiency of parallel reasoning.

2. ALGORITHM'S DATA STRUCTURE AND FORMAL DESCRIPTION

Assume the parallel system contains n processors, which can be called as P_1, P_2, \dots, P_n respectively, in order to realize the parallel reasoning, a queue-stack Q_i ($i=1, 2, \dots, n$) should be created for each processor P_i ($i=1, 2, \dots, n$). These queue-stacks are used to store the generated nodes in the reasoning process, and usually each generated child node is pushed into the top of the same queue-stack. Then using quick sorting algorithm QUICKSORT, we sort these nodes according to their certainty fact, which puts the nodes with higher certainty fact on the top of the queue-stack. When a load-balancing is carried out, the processor with lighter load moves some nodes from the bottom of the queue-stack of the processor with heavier load and puts them on the top of its own queue-stack. We also assume there is a common array used to record the number of nodes in the queue-stack of every processor.

The following is the formal description of our parallel reasoning algorithm:

```

Procedure Parallel Reasoning (T)
begin
  if i=1 then
    begin
      PARENT(T) ← 0;
    end
  end
end

```

```

if T contains goal states then
    PRINT(T)
else
    ANS ← 0;
    /*Assign initial values to three signal variables */
    X1 ← 1; X2 ← 1; X ← 1;
    /* Set queue-stack SQ1 to be empty */
    S1 ← 0; R1 ← 0;
    SQ1[1] ← T; S1 ← 1; R1 ← 1;
end;
FINISH ← 0;
for each Pi (i=1,2,...,n) par do
    while FINISH = 1 do
        begin
            while Si = 0 do
                begin
                    P(X1);
                    /* Pop a node out of the queue-stack SQi */
                    E ← SQi[Si];
                    Si ← Si - 1;
                    V(X1);
                    S0 = Si;
                    /* Search for matching rules from the
                    rule-base ,then use the found rules to extend the
                    current node to produce all the succeeding nodes */
                    SEARCH_MATCH();
                    for each child Ej of E
                        if Ej contains goal states then
                            begin
                                P(X2);
                                ANS ← Ej;
                                while ANS = 0 do
                                    begin
                                        PRINT(ANS);
                                        ANS ← PARENT(ANS)
                                    end;
                                V(X2);
                            end
                            /*If Ej's certainty fact Cmin ,it
                            can be added to the queue-stack */
                        else if CF(Ej) > Cmin then
                            if depth(Ej) < Dmax then
                                begin
                                    P(X1);
                                    /* Put a node into
                                    the queue-stack */
                                    SQi(Si+1) ← Ej;
                                    Si ← Si + 1;
                                    V(X1);
                                end
                            end
                    end
                end
            end
            /*if the search depth go beyond the limit
            then the user is asked about whether
            to enlarge the limit*/
            inquiry();
            P(X3);
            /*Calculate the number of nodes in the
            queue-stack SQi and assign it to A[i]*/
            A[i] ← Si - Ri + 1;
            V(X3);
            /*Sort the newly added nodes by quick sorting
            algorithm, put the nodes with larger certainty
            fact on the top of the queue-stack */
            QUICKSORT (SQi, S0, Si)
        end {while Si = 0};
    end {for each Pi};

```

```

if Si = 0 then
    /*If the queue-stack is empty then
    algortim is called to balance the loads */
    LOAD_BALANCING(i)
end {while FINISH = 1}
end {of procedure}.

```

In the above algorithm ,LOAD_BALANCING() is an algorithm used to balance the loads of all processors. That is, in order to maximize processor utilization during the search phase, a load balancing phase is invoked whenever one or more processors exhaust their stacks and at least one stack contains multiple nodes.

```

Procedure LOAD_BALANCING (i:integer)
begin
    /* First calculate the total of
    nodes in all queue-stacks */
    total ← 0;
    for j ← 1 to n do
        total ← total + A[j];
    if total = 0 then FINISH ← 1;
    else
        begin
            L ← [total/n];
            P(X3);
            K ← 1
            while A[i] < L do
                if A[k] > L then
                    begin
                        /*Move the rear nodes of Pk
                        to the queue-stack of Pi,and
                        adjust the rear pointer of the
                        queue-stack of Pk */
                        NODE_MOVE(k,i);
                        A[i] ← A[i] + 1;
                        A[k] ← A[k] - 1
                    end
                end
            else
                k ← k + 1;
            V(X3);
        end
    end {of Procedure LOAD_BALANCING }.

```

3. ALGORITHM'S ANALYSIS AND DISCUSSIONS

3.1 About Algorithm Parallel-Reasoning

Many problems in artificial intelligence are so complicated that they cannot be solved directly; hence they resort to search technology. But most search methods cannot solve combinatorial explosion problem in the searching process, so they are referred to as "weak methods" in artificial intelligence. In order to avoid the combinatorial explosion in the problem-solving process, heuristic information should be added to the search algorithm, in this way it cost less to find the solutions in most cases. But it cannot ensure that a solution can be found in any case, this is another manifestation of "weak method". Of course, if "strong" heuristic information has been added to the search algorithm, the problem-solving process can show its "strong" effect.

If we had not used ingenious heuristic function in our search algorithm, it would have had exponential run time($\sim 1.2e^N$,

where N is the scale of problem) just as Branch and Bound algorithm does. But fortunately, we have induced a certainty-factor function as heuristic function into our search algorithm, which reduces greatly the problem-solving cost of the algorithm, and especially with parallelization of reasoning research and the application of an efficient dynamic load balancing procedure, the performance of our algorithm has been improved significantly.

3.2 About Algorithm LOAD-BALANCING

Load balancing is a measure of the time spent on processor communication and has a large effect on performance of a parallel algorithm since a sequential algorithm requires no such communication. The main measure of load balancing is simply the fraction of time spent load balancing with respect to the total execution time of the algorithm and is given as $F_L(P) = T_L(P)/T(P)$, Where P is the number of processors used, $T_L(P)$ is the total amount of time spent on load balancing and $T(P)$ is the total execution time.

We also measured two other aspects of load balancing-average load balancing time and load balancing frequency. The average load balancing time is simply the total load balancing time divided by the number of load balancing cycles. The load balancing frequency is measured with respect to node generation and is given as the percentage of the number of node generation cycles after which load balancing is invoked and performed.

Experimental results show that the average time for a load balancing cycle using algorithm LOAD_BALANCING increases with problem size, and that the load balancing frequency for algorithm LOAD_BALANCING is very low and approaches very close to zero for larger problems.

4. REFERENCES

- [1] Ambuj Mahanti and Charles J.Daniels, A SIMD Approach to Parallel Heuristic Search, Artif. Intell. 60(2) (1993) 243-282.
- [2] Chen Guoliang, Parallel Computation---Structure, Algorithm and Programming, Higher Education Press, 1999.5.
- [3] Chen Huaping and etc, A Parallel Realization of Parallel Reasoning on MIMD Model, Minicomputer and Microcomputer, Vol.17, No.4, 1996.
- [4] C.Powley, C.Ferguson and R.E.Korf, Depth-first heuristic search on a SIMD machine, Artif. Intell. 60(2) (1993) 199-242.
- [5] Minghe Huang and Cuixiang Zhong, "A parallel processing method of divide-and-conquer and a Highly efficient parallel sorting algorithm", Proceedings of the annual meeting of Chinese for Theoretic Computer Science in October, 1998, 86-88, in Chinese.



Minghe Huang is a Full professor and a dean of software college in Jiangxi Normal University, a member of the CPPCC Jiangxi Province Committee, the young and middle-aged academic director in Jiangxi Province. He graduated from Jiangxi Normal University in January of 1982, and then in 1986 he attended advanced studies in the department of computer science of Fudan University. His research interests are in distributed parallel processing,

algorithm design and analysis, object-oriented modeling technology, etc. He has published several books and 50 more papers on magazines such as "computer research and development", "computer science", "computer engineering and science", etc.



Cuixiang Zhong is a lecture and an engineer of software college in Jiangxi Normal University. He graduated from computer science and Technology Department of Peking University and got a B.S. in computer science in 1986. He was a visiting scholar to U.B.C in Canada from 1990. He also graduated from Computer science and Technology Department of Jiangxi Normal University and got a M.S. in computer science in 1998. His research interests are in distributed parallel processing algorithm design and analysis, object-oriented modeling technology, e-commerce, MIS, real-time monitoring system. He has published 20 more papers.

A Fast and Efficient Parallel Sorting Algorithm on LARPBS

Chen Hongjian¹ Chen Yixin² Chen Ling¹ Li Tu¹

¹ Department of Computer Science, Yangzhou University, Yangzhou 225009

² Department of Computer Science Univ. of Illinois at Urbana-Champaign, Urbana, IL, 61801, U.S.A

Email: yzchj@yzcn.net

Tel: +86(0514) 7872681

ABSTRACT

A scalable fast parallel sorting algorithm on linear array with reconfigurable pipeline optical bus system (LARPBS) is presented. The algorithm improves Y. Pan's fast parallel sorting algorithm on LARPBS which uses N processors to sort N elements in average $O(N)$ time or optimally $O(\log N)$ time. We illustrate the algorithm can sort N elements in $O(N \log N/p)$ time in the best case and in $O(N^2/p)$ in the worst case using p ($p \leq N$) processors and hence show the algorithm is highly scalable. We also present a fast and efficient parallel sorting algorithm on LARPBS which uses N processors in $O(\log \sqrt{N})$ time in the best case and $O(\sqrt{N})$ time in the worst case.

Keywords: LARPBS model, scalable, sorting, parallel algorithm

Sorting is an important problem in computer science and is widely used in many application fields. Most of the existing parallel sorting algorithm are based on the PRAM model. However, large-scale parallel computation on shared memory system is impractical at least using the current technology. As in the distributed memory system, efficient communication is the most difficult issue in parallel algorithm designing. A high performance parallel algorithm requires architecture and technology that support high communication bandwidth. One way to overcome this difficulty is to use electric/optical buses for communication since they provide direct connection any two processors in the system. Recently, many researchers present some architectures with optical buses such as Reconfigurable Optical Buses (AROB)[1], Reconfigurable Arrays with Spanning Optical Buses (RASOB)[2] and linear array with reconfigurable pipeline optical bus system (LARPBS)[3,4]. In these structures, lots of parallel sorting algorithms were presented. Based on quick sorting algorithm, Y. Pan presented a parallel sorting algorithm on LARPBS which uses N processors to sort N elements in $O(\log N)$ time in the best case and $O(N)$ in the worst case. Subsequently, Yi. Pan et. Al. presented a parallel sorting algorithm on LARPBS using N^2 processors to sort N elements in $O(1)$ time.

While speed is an important motivation for parallel computing, there is another issue in realistic parallel computing, namely, scalability, which measures the ability to maintain speedup linearly proportional to the number of processors. A parallel algorithm must be scalable for the purpose of practical and high performance. The parallel algorithm designers always assume that the scale of parallel computer running the algorithm varies with the problems to be solved. In fact, it is not the case. The number of the processors in an actual parallel computers is fixed while the scale of problems to solve is always variable. Hence, the research of scalability is quite important. We say that a parallel algorithm is scalable in the range $[p_1, p_2]$, if linear speedup can be achieved for all $p_1 \leq p \leq p_2$, where p is the number of processors used. In the other words, suppose the

time complexity of a parallel algorithm be presented as $O(T(n))$ using p processors where n is the size of the problem, for some constant $r > 0$, if p/r processors are used, the time complexity must be bounded by $rO(T(n))$ in order to be scalable. In this paper, a scalable fast parallel sorting algorithm on linear array with reconfigurable pipeline optical bus system (LARPBS) is presented. The algorithm improves Y. Pan's fast parallel sorting algorithm on LARPBS which uses N processors to sort N elements in average $O(N)$ time or optimally $O(\log N)$ time. We also illustrate the algorithm can sort N elements in $O(N \log N/p)$ time in the best case and in $O(N^2/p)$ in the worst case using p ($p \leq n$) processors and hence show the algorithm is highly scalable. Finally, we present a fast and efficient parallel sorting algorithm on LARPBS which uses N processors in $O(\log \sqrt{N})$ time in the best case and $O(\sqrt{N})$ time in the worst case.

1. THE LARPBS MODEL

A pipelined optical bus system uses optical waveguides instead of electrical buses to transfer messages among electronic processors. The advantages of using waveguides can be seen as follows: Besides of the high propagation speed of light, there are two important properties of optical signal (pulse) transmission: high bandwidth and predictable message delay. These two properties enable synchronized concurrent access of an optical bus in a pipelined fashion and efficient broadcasting or multicasting of the bus structure, which make the architecture suitable for many applications that involve intensive communication operations.

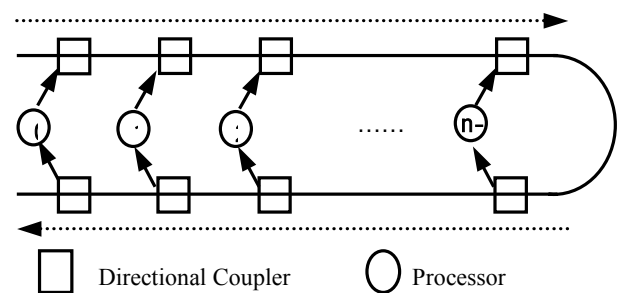


Figure 1 A linear optical bus system of N processors

Figure 1 shows a linear array in which electronic processors are connected with an optical bus. Each processor is connected to the bus with two directional couplers, one for transmitting on the upper segment and the other for receiving from the lower segment of the bus. The optical bus contains three identical waveguides, one for carrying messages (the message waveguide) and the other two for carrying address information (the reference waveguide and the select waveguide), as shown in Figure 2. For the purpose of simplicity, the message waveguide, which resembles the reference waveguide, has been omitted from the figure. Messages are organized as fixed-length message frames. Note that optical signals propagate

unidirectionally from left to right on the upper segment and from right to left on the lower segment. This bus system is also referred to as the folded-bus connection in [1]

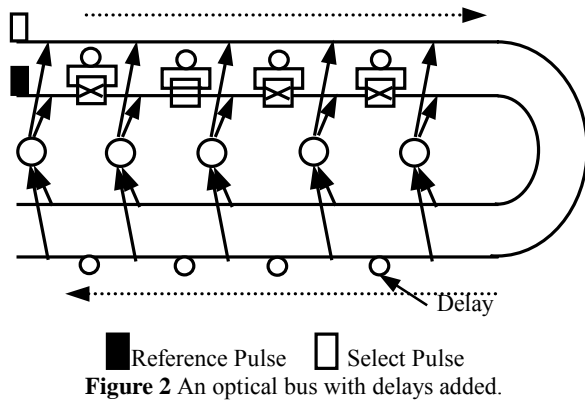


Figure 2 An optical bus with delays added.

Let ω be the pulse duration in seconds and C_b the velocity of light in these waveguides. Define a unit delay Δ to be the spatial length of a single optical pulse, that is $\Delta = \omega \cdot C_b$. Initially, processors are connected to these three waveguides such that between any two given processors, the same length of fiber is used on all three waveguides. Hence, the propagation delays between two processors are the same for all three waveguides. A bus cycle for an optical bus is defined as the end-to-end propagation delay on the bus; i.e., the time taken for an optical signal to propagate through the entire bus. If T is the time taken for a signal to traverse the optical distance between two consecutive processors on the bus, then the length of a bus cycle for the system of figure 3 is $2N \cdot T$. We then add one unit delay Δ (shown as a loop in fig.2) between any two processors on the receiving segments of the reference waveguides and of the message waveguides. Each loop is an extra segment of a fiber and the amount of delay added could be accurately chosen based on the length of the segment. As a result, the propagation delays on the receiving segments of the select waveguide and the reference waveguides are no longer the same. Finally, we add a conditional delay Δ between any two processors $i+1$ and i where $0 \leq i \leq N-2$, on the transmitting segments of the select waveguides. The switch between processor $i+1$ and i is called $S(i+1)$ and is local to processor $i+1$. Thus, every processor has its own switch, except processor 0. Each switch can be set by the local processor to two different states: straight or cross. When a switch is set to straight, it takes T time for an optical signal on the transmitting segments of the select waveguides to propagate from one processor to its nearest neighbor. When a switch is set to cross, a delay is introduced and such propagation will take $T + \omega$ time. Clearly, the maximum delay that the switches can introduce is the duration of $N-1$ pulses.

Messages transmitted by different processors may overlap with each other even if they propagate unidirectionally on the bus. We call these message overlapping transmission conflicts. Assume each message has b binary bits, each bit represented by an optical pulse, with the existence of a pulse for 1 and the absence for 0. To ensure that there are no transmission conflicts, the following condition must be satisfied: $T > \omega b$, where T is the time taken for a signal to traverse the optical distance between two consecutive processors on the bus, and ω is the pulse duration. Note that the above condition ensures that each message can fit into a pipeline cycle such that in a bus cycle, up to N messages can be transmitted by processors simultaneously without collisions on the bus. In a parallel array, messages

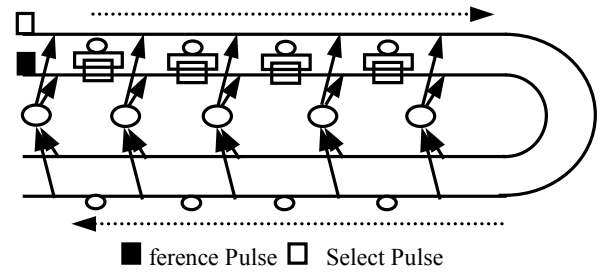


Fig 3 Switch settings for a broadcast operation

normally have very short length; i.e., b is very small. Thus, in the following discussion, we assume that the above condition is always satisfied and that no transmission conflicts are possible as long as all processors are synchronized at the beginning of each bus cycle.

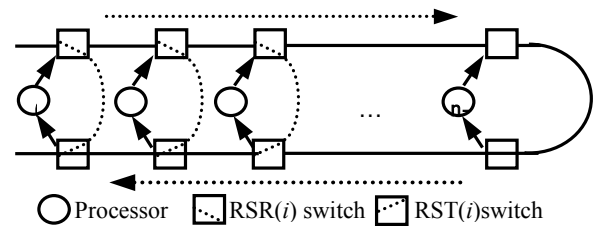


Fig 4 The LARPBS model of size n with two subarrays

In the LARPBS, we insert an optical switch on each section of the transmitting bus and receiving bus. Thus, each processor has 6 more local switches, three on its three receiving segments and three on its three transmitting segments, besides its switch for conditional delay. The switches on the receiving and transmitting segments between processors i and $i+1$ are called $RSR(i)$ and $RST(i)$, respectively, and are local to processor i as shown in Figure 4. Here, $RSR(i), 0 \leq i \leq N-2$ are 2×1 optical switches, and $RST(i)$ are 1×2 optical switches. In the following discussion, these switches will be called reconfigurable switches due to their function. When all switches are set to straight, the bus system operates as a regular pipelined bus system. When $RSR(i)$ and $RST(i)$ are set to cross, the whole bus system is split into two separate systems, one consisting of processors $0, 1, \dots, i$ and the other consisting of $i+1, i+2, \dots, N-1$. The total delay for a signal passing the transmitting segment, the optical fiber between $RST(i)$ and $RSR(i)$, and the receiving segment is made to be equal to T . Here, the array with processors 0 to i can operate as a regular linear array with a pipelined bus system: so does the array with processors $i+1$ to $N-1$. Figure 4 shows the LARPBS model with N processors. The array is split into two subarrays, with the first one having $i+1$ processors and the second one having $N-i-1$ processors. In the figure, only one waveguide is shown. Conditional switches are omitted in the figure to avoid confusion.

2. BASIC DATA MOVEMENT OPERATIONS ON LARPBS

The following primitive operations on LARPBS have been used extensively in parallel algorithm design on LARPBS. Our algorithms are developed by using these operations as building blocks.

One-to-One Communication Assume that processors

$P_{i_1}, P_{i_2}, \dots, P_{i_m}$ are senders, and processors $P_{j_1}, P_{j_2}, \dots, P_{j_m}$ are receivers. In particular, processor P_{i_k} sends its value in its register $R(i_k)$ to the register $R(j_k)$ in P_{j_k} . The operation is represented as

for 1 $k \leq m$ par-do $R(j_k) \leftarrow R(i_k)$.
(Note that we use $R(i)$ to denote both the name and the content of register $R(i)$.)

Broadcasting Here, we have a source processor P_i , who sends a value in its register $R(i)$ to all the N processors:

$R(i_1), R(i_2), \dots, R(i_m) \leftarrow R(i)$.

Multiple Multicasting A multiple multicasting operation is denoted as follows:

for 1 $k \leq g$ par-do
 $R(j_{k,1}), R(j_{k,2}), R(j_{k,3}), \dots, R(i_k)$.

Element Pair-Wise Operations The operation is described as follows:

for 1 $k \leq N^2$ par-do
 $R(m+k-1) \leftarrow R(m+k-1) \oplus R(n+k-1)$.

Compression Compression operation is represented as

for 1 $k \leq m$ par-do $R(N-m+k-1) \leftarrow R(i_k)$.

Binary Prefix Sums The operation is represent as

for 0 $j \leq N-1$ par-do $R(j) \leftarrow R(0) + R(1) + \dots + R(j)$.

Binary Value Aggregation The operation is represent as

$R(0) \leftarrow R(0) + R(1) + \dots + R(N-1)$.

The reader is referred to [4] for implementation details of these operations on optical buses, where the following results are established.

In the LARPBS computing model, one-to-one communication, broadcasting, multiple multicasting, element pair-wise operation, binary prefix sums and compression operation, binary value aggregation take $O(1)$ time.

3. A SORTING ALGORITHM USING N PROCESSORS IN $O(\log N)$ TIME IN THE BEST CASE AND $O(N)$ IN THE WORST CASE

The parallel sorting algorithm can be described by a recursive procedure $\text{PARSORT}(l, u)$ which sorts the elements in the subarray (l, u) , where l is the starting number and u is the terminal number of the processors in the array. Assume that element a_i is stored in the memory unit $R[i](0 \leq i \leq N-1)$ in P_i ($i=1, 1, \dots, N$). The procedure $\text{PARSORT}(l, u)$ can be described as follows:

PROCEDURE $\text{PARSORT}(l, u)$

```
{if  $l=u$  then end
else
{Step 1 For  $l \leq i \leq u$  par-do  $F[i] \leftarrow R[i]$ 
Step 2 For  $l \leq i \leq u$  par-do
    if  $R[i] > F[i]$  then  $S[i] \leftarrow 1$  else  $S[i] \leftarrow 0$ 
Step 3 For  $l \leq i \leq u$  par-do
     $T[i] \leftarrow S[l] + S[l+1] + \dots + S[i]$ 
     $m \leftarrow T[u]$ ;
Step 4 For  $l \leq i \leq u$  par-do
    if  $S[i]=1$  then  $F[l+T[i]-1] \leftarrow R[i]$ 
```

Step 5 For $l \leq i \leq u$ par-do $S[i] \leftarrow \overline{S[i]}$;

For $l \leq i \leq u$ par-do

$T[i] \leftarrow S[l] + S[l+1] + \dots + S[i]$;

Step 6 For $l \leq i \leq u$ par-do

if $S[i]=1$ then $F[l-1+m+T[i]] \leftarrow R[i]$

Step 7 For $l \leq i \leq u$ par-do $R[i] \leftarrow F[i]$;

Step 8 do $\text{PARSORT}(l, l+m-1)$ and

$\text{PARSORT}(l+m+1, u)$ in parallel ;

}}

The algorithm is just the process of $\text{PARSORT}(0, N-1)$.

It can be easily seen that the algorithm uses only N processors. Step1 to step7 of the algorithm are all primitive operations on LARPBS which can be finished in $O(1)$ time. Step8 is a recursion operation. Therefore the time complexity of the algorithm is just the recursion depth of procedure $\text{PARSORT}(0, N-1)$ which is obviously $O(\log N)$ in the best case and $O(N)$ in the worst case. Consequently, we can get the following conclusion: On LARPBS model, we can use N processors to sort N elements in $O(\log N)$ time in the best case and $O(N)$ in the worst case, and the cost in $O(N^2)$ in the worst case.

4. A FAST SCALABLE PARALLEL ALGORITHM USING p PROCESSORS

Now we show that our algorithm is highly scalable. When the LARPBS model contains $p(1 \leq p \leq N)$ processors, we can divide N elements into p groups each one containing N/p elements (generally assuming N/p is an integer). So the whole sequence can be treated as a two-dimension $N/p \times p$ matrix. We can place the k th elements in the r th processor as the s th element, where $r = \lfloor k/q \rfloor$, $s = k \bmod q$. For the elements in $N/p \times p$ matrix, we can use rotate sorting [5,6] presented by Maberg and Gafni to realize the scalable parallel sorting. Three subroutines named BALANCE, UNLOCK and SHARE are used in the algorithm. The implementation details of these subroutines are shown as follows:

PROCEDURE BALANCE (v, w)

{Step 1 For 0 $k \leq N/p-1$ do using $\text{PARSORT}(0, p-1)$ to sort all elements of row k collaterally;

Step 2 Rotate the elements of processor P_i , $i \bmod v$ positions downward;

Step 3 For 0 $k \leq N/p-1$ do using $\text{PARSORT}(0, p-1)$ to sort all elements of row k collaterally;}

PROCEDURE UNBLOCK

{Step 1 Rotate the elements of processor P_i , $i \cdot \sqrt{N/p} \bmod N/p$ positions downward;

Step 2 For 0 $k \leq N/p-1$ do using $\text{PARSORT}(0, p-1)$ to sort all elements of row k ;

PROCEDURE SHEAR

{Step 1 Merging sort all elements of even numbered processors upward collaterally and all elements of odd numbered processors downward collaterally;

Step 2 For 0 $k \leq N/p-1$ do using $\text{PARSORT}(0, p-1)$ to sort all elements of row k }

PROCEDURE PARSORT1

{Step 1 Do BALANCE($N/p, \sqrt{N/p}$) for each column of the submesh;
 Step 2 Do UNBLOCK for entire mesh;
 Step 3 Do BALANCE($\sqrt{N/p}, p$) for each row of the submesh;
 Step 4 Do UNBLOCK for entire mesh;
 Step 5 For 0 $k \leq 2$ do SHEAR;
 Step 6 Merging sort the elements of each processor upward serially;}

The complexity analysis of the algorithm is as follows: BALANCE, PARSORT can use p processors to finish row sorting in $O(p)$ time in the worst case and $O(\log p)$ in the best case. N/p rows can be processed during the time from $N/pO(\log p)$ to $N/pO(p) = O(N) \sim O(N^2)$. The rotate operation of N/p elements can be finished in $O(N/p)$ time. So the time complexity of the whole process is $N/pO(\log p)$ in the best case and $O(N) \sim O(N^2/p)$ in the worst case. With the same reason, the time complexity of UNBLOCK is $N/pO(\log p)$ in the best case and $O(N) \sim O(N^2/p)$ in the worst case. For SHARE, it can finish sequential merging sorting of N/p elements in $N/pO(\log N/p)$ time. Hence, the time complexity of SHARE is $\max(N/pO(\log p), N/pO(\log N/p)) \sim N/pO(\log N)$. Because algorithm PARSORT1 adopts the fundamental procedure above to finish sorting, we can come to a conclusion based on algorithm PARSORT: On LARPBS model, N elements can be sorted using p processors in $O(M \log N/p)$ time in the best case and $O(N^2/p)$ in the worst case. Therefore, algorithm PARSORT1 is highly scalable.

5. AN EFFICIENT PARALLEL SORTING ALGORITHM USING N PROCESSORS

Suppose the LARPBS model contains N processors, place N elements into N processors each processing one element. Meanwhile, divide the processor array $(0:N-1)$ into \sqrt{N} subarrays each one containing \sqrt{N} processors. Thus the array can be seen as $\sqrt{N} \times \sqrt{N}$ two-dimensional mesh. Similarly, the rotate sorting presented by Maberg and Gafni can realize to improve parallel sorting. The algorithm uses four subroutines named COLUMNSORT, BALANCE, UNBLOCK1 and SHEAR. The implementation details of these subroutines are shown as follows:

PROCEDURE COLUMNSORT

{Step 1 For $k=0$ to $N-1$ par-do $\{r = \lfloor k / \sqrt{N} \rfloor; s = k \bmod \sqrt{N};$ Broadcast the value of $R[k]$ to $R[s\sqrt{N} + r]\}$
 Step 2 Divide array $\Pi(0 : N-1)$ into \sqrt{N} subarrays and using PARSORT to sort \sqrt{N} elements of each group in parallel;
 Step 3 After sorting, according to the values of r and s , broadcast the elements to their original processors}

PROCEDURE BALANCE1(v, w)

{Step 1 COLUMNSORT
 Step 2 For $k=0$ to $\sqrt{N}-1$ par-do rotate the elements of $\Pi(k\sqrt{N} : (k+1)\sqrt{N}-1), k \bmod w$ positions right;
 Step 3 COLUMNSORT}

PROCEDURE UNBLOCK1

{Step 1 For $k=0$ to $\sqrt{N}-1$ par-do rotate the elements of $\Pi(k\sqrt{N} : (k+1)\sqrt{N}-1), k \cdot \sqrt{N} \bmod \sqrt{N}$ positions right;
 Step 2 COLUMNSORT}

PROCEDURE SHEAR1

{Step 1 For $k=0$ to $\sqrt{N}-1$ par-do sort the elements of even numbered subarray $\Pi(k\sqrt{N} : (k+1)\sqrt{N}-1)$ upward and the elements of odd numbered subarray $\Pi(k\sqrt{N} : (k+1)\sqrt{N}-1)$ downward;
 Step 2 COLUMNSORT}

PROCEDURE PARSORT2

{Step 1 Do BALANCE(\sqrt{N}, \sqrt{N}) for each row of the block;
 Step 2 Do UNBLOCK for the entire array;
 Step 3 Do BALANCE(\sqrt{N}, \sqrt{N}) for each column of the block;
 Step 4 Do UNBLOCK for entire array;
 Step 5 For 0 $k \leq 2$ do SHEAR
 Step 6 Sort the elements of each subarray upward;}

The complexity of the algorithm is shown as follows: On COLUMNSORT, step1 and step3 can be finished in $O(1)$ time. According to the analysis of PARSORT, step2 can be finished in $O(\log \sqrt{N})$ time in the best case and $O(\sqrt{N})$ in the worst case using \sqrt{N} processors to sort \sqrt{N} elements. For BALANCE1, recycle rotate can perform by the primitive operation of multicasting in $O(1)$ time. Hence, the time complexity of the BALANCE1 decided by COLUMNSORT is $O(\log \sqrt{N})$ in the best case and $O(\sqrt{N})$ in the worst case. Hence, the time complexity of PARSORT4 is $O(\log \sqrt{N})$ in the best case and $O(\sqrt{N})$ in the worst case. Subsequently, we come to the conclusion: On LARPBS, using N processors to sort N elements, the time complexity is $O(\log \sqrt{N})$ in the best case, and $O(\sqrt{N})$ in the worst case. The cost is $O(N^{3/2})$ in the worst case. Obviously, performance of Algorithm PARSORT1 is superior to that of algorithm PARSORT2.

6. CONCLUSIONS

Linear array with reconfigurable pipeline optical bus system (LARPBS) is an efficient parallel computing model. It provides an important foundation for the research and the development of parallel computing architecture using optical interconnection. It has wide applications in many fields. In this paper, a scalable fast parallel sorting algorithm on linear array with reconfigurable pipeline optical bus system (LARPBS) is presented. The algorithm improves Y. Pan's fast parallel sorting algorithm on LARPBS which uses N processors to sort N elements in average $O(N)$ time or optimally $O(\log N)$ time. We illustrate the algorithm can sort N elements in $O(M \log N/p)$ time in the best case and in $O(N^2/p)$ in the worst case using p ($p \leq n$) processors and hence show the algorithm is highly scalable.. We also present a fast and efficient parallel sorting algorithm on LARPBS which uses N processors in $O(\log \sqrt{N})$ time in the best case and $O(\sqrt{N})$ time in the worst case.

Although the cost of our second algorithm is $O(N^{3/2})$ in the

worst case, it is still better than that of Y.Pan's. Our algorithm takes advantage of many important merits of LARPBS such as its high communication bandwidth, the versatile communication patterns it supports, and its ability of utilizing communication reconfigurability as an integral part of a parallel computation. It is shown that the LARPBS is a powerful architecture for exploiting large degree of parallelism in a computational problem that most other machine models cannot achieve. Recently, some parallel algorithms on LARPBS have been researched and designed such as matrix calculation, sorting, selecting, merging and diagram theory. But the parallel algorithm and scalability of many application problems need further researching and improving. Moreover, the research on LARPBS model and parallel algorithm can improve the optical interconnection technology and parallel processing technology. It is significant to the realization of parallel computers with optical interconnections and the development of high quality computers.

7. REFERENCES

- [1] S.Pavel,S.G. Akl, Computing the Hough transformation on arrays with reconfigurable optical buses, in *Parallel Computing Using Optical Interconnections*, K. Li, Y. Pan, and S. Q. Zheng, eds., Kluwer Academic Publishers, Boston, USA, Hardbound, ISBN 0-7923-8296-X, October 1998, pp205-226 .
- [2] C.Qiao, Y. Mei, On Efficient Embedding of Binary Trees in Reconfigurable Arrays with Spanning Optical Buses, *Int'l Journal on Parallel and Distributed Systems and Networks*, Vol. 2, No. 1, pp. 40-48, 1999
- [3] Y. Pan and K. Li, " Linear array with a reconfigurable pipelined bus system -- concepts and applications," *Information Sciences* , Vol. 106, No. 3/4, May 1998, pp. 237-258.
- [4] Y. Pan, "Basic data movement operations on the LARPBS model " , in *Parallel Computing Using Optical Interconnections*, K. Li, Y. Pan, and S. Q. Zheng, eds., Kluwer Academic Publishers, Boston, USA, Hardbound, ISBN 0-7923-8296-X, October 1998 , pp227-247.
- [5] J.M.Maberg and E.Gafni, Sorting in constant number of row and column phases on a mesh.*Algorithmica* 3(4),1988,pp561-572.
- [6] M. Hamdi, J. Tong, C.W. Kin, Fast sorting algorithms on reconfigurable array of processors with optical buses, 1996 International Conference on Parallel and Distributed Systems,1996,pp183-186.

Fast Parallel Identification of Multi-peaks in Function Optimization *

Guo Guanqi, Tan Zhumei

Department of Computer and Information Engineering, Hunan Institute of Science and Technology

Yueyang, Hunan 414000, China

Email: guanqi_guo@hotmail.com Tel.: +86(0)730-8844171

ABSTRACT

A class of hybrid niching evolutionary algorithms (HNE) using clustering crowding and parallel local search is proposed. By analyzing topology of fitness landscape and extending the space for searching similar individual, HNE determine the locality of search space more accurately, thus decreasing the replacement errors of crowding and suppressing genetic drift of the population. The integration of deterministic and probabilistic crowding increases the capacity of both parallel local hill-climbing and maintaining multiple subpopulations. Parallel local search based on simplex method over disjoint subpopulations greatly speed up the convergence of the population towards various optima simultaneously. Real coded representation and Gaussian mutation improve the precision of the solutions founded. The experimental results optimizing various multimodal functions show that, the performances of HNE such as the number of effective peaks generated and maintained, average peak ratio, global optimum ratio and CPU time consumed are uniformly superior to those of genetic algorithms using sharing, deterministic crowding method.

Keywords: Evolutionary Algorithms, Genetic Drift, Niche, Clustering Crowding, Parallel Local Search.

1. INTRODUCTION

Genetic Algorithms [1] (GA) are a class of search and optimization techniques modeled from organic evolution and population genetics. It is well known for its global exploration, implicit parallelism and robustness to a wide range of problems. However, genetic drift and slow convergence velocity may prevent a GA from being really of practical interest for a lot of applications, especially when convergence rate, reliability and multiple solutions are simultaneously required.

Niching methods provide GA with the capacity suppressing genetic drift of the population. Fitness share [2] (SH) is a best well known niching technique. It introduces explicit restoration pressure of species in GA to maintain stable subpopulations. But the parameterization of niche radius requires a prior knowledge of the problem, which is unavailable in some situations. Deterministic crowding [3] and probabilistic crowding [4] are simple niching methods computation cost efficient and parameter irrelative, they can be easily applied to a wide range of problems. However, it is not able to overcome genetic drift effectively because of a large number of replacement errors of similar individuals, especially as a small population size is adopted.

This paper investigates a hybrid niching evolutionary

algorithm (HNE) for multimodal function optimization. HNE incorporates clustering crowding and parallel local search the authors proposed in [5] to identify multiple global or local optima and speed up search simultaneously. The detailed descriptions of the proposed algorithm, performance criterions, and experiment results are presented in the following sections.

2. FAST PARALLEL IDENTIFICATION METHODS

Representation

As indicated in [6], the coding function of the binary representation might make the search procedure more complex than the original problem was. Suppose the population size is N , each solution is encoded as a string of l bits. Decoding of binary strings and mutation introduce additional computational costs in the order of $O(N \times l)$, which take a non-trivial part in the total CPU time required for running a GA. Moreover, a standard GA is not able to tune the precision of the resulting solution with effective computation costs. In order to overcome these drawbacks, HNE uses real coded representation. For an n -dimensional function optimization problem, the i th individual in a population is directly represented by the objective variable $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$, where, real number (float point number) $x_{ij} (j=1, \dots, n)$ is called a gene, which denotes the j th component of \mathbf{x}_i .

Gaussian Mutation

Suppose the domain of an n -dimensional optimization problem is $[\mathbf{u}, \mathbf{v}] = \prod_{i=1}^n [u_i, v_i]$. The individual $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{in})$ is mutated using the equation

$$x'_{ij} = x_{ij} + \sigma \cdot N_j(0, 1), \quad j = [1, \Lambda, n] \quad (1)$$

Where, $N_j(0, 1)$ is a normally distributed random number with expectation zero and standard deviation σ . The index j indicates that the random variable is sampled anew for each component of the objective variable. σ is called step size. It is dynamically modified according to a simple adaptive criterion as follows.

Let σ and σ' respectively indicates the current step size and the next one, and $\lambda \geq 1$ denotes the adaptive learning rate. If all mutations during successive 5 generations do not improve the best-so-far individual, σ is set to value of $\sigma' = \sigma/\lambda$. If the mutations in each of successive 5 generations improve the best-so-far individual, $\sigma = \sigma \cdot \lambda$. Otherwise, the step size remains unchanged.

Certainly, the optimal settings of the initial step size σ_0 and the learning rate λ are not completely independent of optimization problems, but our empirical data showed that the values of $\sigma_0 = (0.01 \sim 1)L$ and $\lambda = 1.1 \sim 3$ were very robust to all problems tested, where,

* Supported by the National Natural Science Foundation of China under Grant No. 50275170; The Science Research Foundation of Education Office of Hunan Province under Grant No. 2002A052.

$$L = \frac{1}{n \times N} \sum_{i=1}^n (v_i - u_i) \quad (2)$$

Recombination

HNE uses two-point crossover, i.e., let

$$\begin{aligned} x_i &= (x_{i1}, \Lambda, x_{i\mu}, \Lambda, x_{iv}, \Lambda, x_{in}) \\ x_j &= (x_{j1}, \Lambda, x_{j\mu}, \Lambda, x_{jv}, \Lambda, x_{jn}) \end{aligned} \quad (3)$$

be two randomly selected parental individuals, where, μ and v respectively indicates the random integer numbers in the interval $[1, n]$, and $\mu \leq v$, the children produced by recombination are

$$\begin{aligned} x'_i &= (x_{i1}, \Lambda, x_{j\mu}, \Lambda, x_{jv}, \Lambda, x_{in}) \\ x'_j &= (x_{j1}, \Lambda, x_{i\mu}, \Lambda, x_{iv}, \Lambda, x_{jn}) \end{aligned} \quad (4)$$

In each generation, N individuals in the population are randomly mated into $N/2$ pairs of parents, each pair produces two children by recombination. Such a recombination strategy performed very well in almost all optimization tasks tested. It is hard to present a strict theory to justify our choice, but it seems intuitive that two-point crossover supports the building block hypothesis [7] and the genetic repair hypothesis [8].

Clustering Crowding and Parallel Local Search

HNE initializes a population P of size N by random. In each generation, N individuals are randomly partitioned into $N/2$ pairs of parent without replacement. Each pair of parent generates two children by recombination and Gaussian mutation. N children form a temporary population P' . For each child P'_i $[1, N]$, HNE first finds out the closest individual P_{jj} $[1, N]$ measured by Euclidean distance, then uses hill-valley function [9] to analyze whether or not P'_i and P_j locate in an identical equivalence class [3] describing the largest neighborhood of a local optimum (simply called class hereafter). If P'_i and P_j are in an identical class and P'_i fitter than P_j , P'_i replaces P_j deterministically. Otherwise, P'_i replaces P_j with probability proportional to its fitness value. After all the replacements, HNE applies parallel local search operator PLS [5] in population P to search multiple local extremes simultaneously. The pseudocode of HNE algorithm is as follows:

Algorithm HNE

Randomly generate a population P of size N ;

While (not fulfill stopping criterion)

$A = \{1, 2, \dots, N\}$; $B = \emptyset$; $P' = \emptyset$;

For ($k=1$; $k \leq N/2$; $k=k+1$)

i =a random integer in set $A-B$; $B=B \cup \{i\}$;

j =a random integer in set $A-B$; $B=B \cup \{j\}$;

(C_i, C_j) =mutation \circ recombination (P_i, P_j);

$P' = P' \cup (C_i, C_j)$;

For ($i=1$; $i \leq N$; $i=i+1$)

Find out the closest P_{jj} $[1, N]$ away from P'_i ;

Analyze whether or not P'_i and P_j in an identical class by hill-valley function;

If (P'_i and P_j in an identical class and $f(P'_i) > f(P_j)$)

P_j replaces P'_i ;

Elseif ($\kappa \cdot f(P'_i) / (f(P'_i) + f(P_j))$)

P'_i replaces P_j ;

Parallel local search in P using PLS;

End of HNE

(7)

Where, \emptyset denotes empty set, $f > 0$ indicates the fitness function of the objective problem. κ denotes a uniformly distributed random variable in the interval $[0, 1]$, and each reference samples a new value.

HNE searches a similar individual of an offspring from the whole parental population and further uses hill-valley function to analyze the locality of both, so as to increase the exactness of similarity judgement. Deterministic replacement speeds up local hill-climbing. Probabilistic replacement not only reduces genetic drift caused by replacement errors, but also provides survival and maintaining probabilities for newly generated classes and classes with lower fitness. HNE reaches the trade-off of generating and maintaining classes. The independent local search in multiple disjoint subpopulations using PLS can effectively speeds up the convergence velocity of the population towards various local extremes of the search space in parallel, but causing no additional genetic drift.

3. PERFORMANCE CRITERIONS

In designing evolutionary algorithms optimizing multimodal functions and multiobjective problems, the number of global or local optima and the precision of the solutions should be taken into consideration simultaneously such that both the reliability and speed of convergence can be evaluated synthetically. The following criterions are used to measure the performances of niching evolutionary algorithms.

Number of Effective Classes Maintained

If an equivalence class of the search space has at least one solution in the population, and the fitness of the solution is not less 80% of the corresponding local optimum, it is considered to be an effective class. The number of effective classes (NEC) denotes the number of local optima found and maintained in the population. NEC is used to evaluate the capacity that a niching evolutionary algorithm generates and maintains multiple local optima in parallel.

Average Peak Ratio

The solution with the maximum fitness in an effective class is considered to be an effective local optimum. The average peak ratio (APR) is the sum of the fitness of the effective local optima maintained in the population divided by the sum of the fitness of all real local optima in search space. It is an average evaluation of the precisions of the local optima identified by a niching evolutionary algorithm.

Global Optimum Ratio

An effective local optimum with the maximum fitness in the population is considered to be an effective global optimum. The global optimum ratio (GOR) is the fitness of the effective global optimum divided by the fitness of the real global optimum in search space. GOR measures the precision of the global optimum in the population.

4. SIMULATIONS

Test Functions

The following four multimodal functions with different difficulty are used the test bed in our simulation optimizations. In literatures, they are widely used as the test bed of niching evolutionary algorithms.

$$f_1(x) = \sum_{i=1}^{10} ((b_i(x - a_i))^2 + c_i)^{-1}, x \in [-10, 10]$$

This function is a representative of 1-dimensional nonlinear functions with uneven distributed local optima. It has 8 non-uniformly distributed maxima in interval [10,10]. The values of the constants a_i , b_i and c_i refer to literature [10]. The global maximum locates at $x^* = 0.68487$ with function value $f_1(x^*) = 14.59265$.

$$f_2(x) = 500 -$$

$$\begin{aligned} & [0.002 + \sum_{j=1}^{24} (1 + j + (x_1 - a(j))^6 + (x_2 - b(j))^6)^{-1}]^{-1}, \\ & -65.536 \leq x_1, x_2 \leq 65.535, \\ & a(j) = 16((j \bmod 5) - 2), b(j) = 16\lfloor j/5 \rfloor - 2 \end{aligned}$$

This function is a two-dimensional function with 25 maxima located at coordinates of $(16i, 16j)$, where i and j represent all integers in $[-2, 2]$. The 25 maxima are all of differing heights, ranging from 476.191 to 499.002. The global optimum occurs at $(-32, 32)$. The other optima form a staircase of spikes to the global optimum.

$$f_3(d, r, h) = \begin{cases} h - \frac{2hd^2}{r^2}, & d < \frac{r}{2} \\ \frac{2h(d-r)^2}{r^2}, & \frac{r}{2} \leq d < r, r \in [0.02, 0.1], h \in [0.1, 1] \\ 0, & \text{otherwise} \end{cases}$$

This function is called Bell function, r is the radius of the cone, h is the height, and d is the Euclidean distance from the center of the cone. With different number of the peaks and different values of r and h , this function can provide changeable complexity. In our experiments, the centers of 30 bells are randomly initialized in a 2-dimensional space with the domain of $x \in [0, 1]$. The radii of the bells are randomly generated with values in $[0.02, 0.1]$, and the heights in $[0.1, 1]$. The maximum point locates at $(0.76, 0.61)$ with function value of 1.

$$f_4(x) = \prod_{i=1}^2 \sum_{j=1}^5 \{j \cos[(j+1)x_i + j]\}, x_i \in [-10, 10]$$

Shubert function f_3 has 18 symmetrically distributed global optima with the minimum values $f_3(x^*) = 186.73093$. The global optima locate at the needlepoints extruding from a large number of local peaks. Using the algorithms with genetic drift, the search is hard to find out all of the global optima.

Test Algorithms

Two niching genetic algorithms using fitness share (SH), deterministic crowding (DC) and our hybrid niching evolutionary algorithm (HNE) are used to optimize the four test functions. All algorithms use population size of 100. With respect to SH and DC, each dimension of objective variables are encoded as a binary string of 30 bits, the mutation and crossover rate are respectively set to 0.001 and 1. The initial step size and the learning rate of HNE are set to $\sigma_0 = L$ (see Eq.2) and $\lambda = 1.1$. The niche radii are set to 1, 10, 0.02 and 0.3 respectively for f_1, f_2, f_3 and f_4 .

Table 1 The performance data of SH, DC and HNE

		NEC	APR	GOR	T
f_1	SH	6.64	0.75	0.98	3.7

optima=8	DC	5.36	0.64	1.00	2.7
	HNE	7.12	0.88	1.00	1
f_2 optima=25	SH	25	0.86	0.99	3.7
	DC	4	0.14	1.00	3.0
	HNE	25	0.99	1.00	1
f_3 optima=30	SH	5.44	0.15	0.87	3.5
	DC	13.83	0.37	0.98	2.8
	HNE	26.76	0.87	1.00	1
f_4 optima=18	SH	8.46	0.42	0.94	4.1
	DC	7.38	0.39	0.99	3.3
	HNE	18	0.98	1.00	1

Observations

All algorithms perform 100 independent runs for each test function. For the relatively simple f_1 and f_2 , each independent run performs 25000 function evaluations. As to f_3 and f_4 , each run performs 50000 evaluations. The experiment results are measured using the arithmetic mean of NEC, APR, GOR and T over 100 runs, where T denotes the CPU time a algorithm consumes for performing the specified evaluations, and for each function, the time HNE consuming is standardized to 1 unit, the others take the times of the unit. All resulting data are listed in Tab.1, where the number of optima for f_4 only counts 18 global optima considering the complexity.

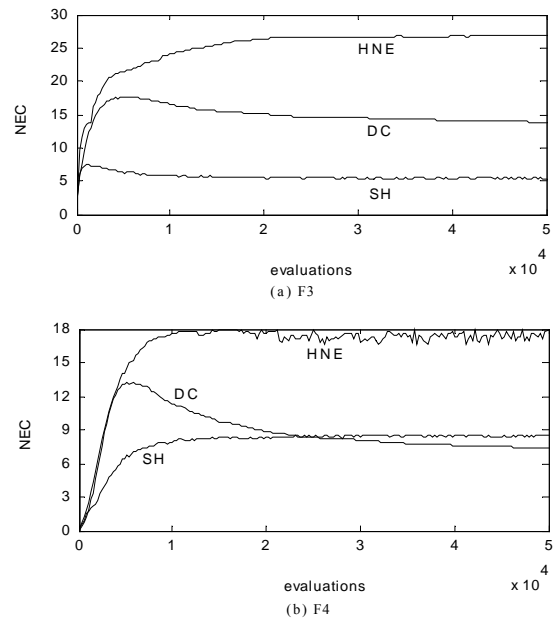


Figure 1 Trend curves of average NEC over 100 runs for optimizing f_3 and f_4

As indicated in Tab.1, for all test functions, HNE finds out almost all the local or global optima. It shows that HNE is able to suppress the genetic drift of the population towards a single local or global optimum efficiently. The values of APR and GOR approximate to 1 and the less CPU time show that HNE can generate solutions with high quality at lower computation cost. Although the clustering analysis using hill-valley function decreases the net number for fitness evaluation, however, the application of PLS remarkably increases the speed of parallel local convergence.

Fig.1 gives the trend curves of average values of NEC over 100 independent runs for optimizing f_3 and f_4 using SH, DC and HNE. It is clearly seen that in initial stage of the search

procedure, HNE shows the fast speed generating new effective classes. The number of effective classes generated and maintained by HNE increases as the search proceeds until reaching equilibrium state. In equilibrium, the value of NEC equals approximately to the number of real optima in the search space.

The diversity of the initial population and the deterministic replacement endow DC with faster local hill-climbing and stronger capacity generating effective classes in earlier stage of the search procedure. As the search proceed, the diversity of the population decreases, the genetic drift resulting from replacement errors overwhelms increasingly the advantage of local hill-climbing from deterministic replacement such that

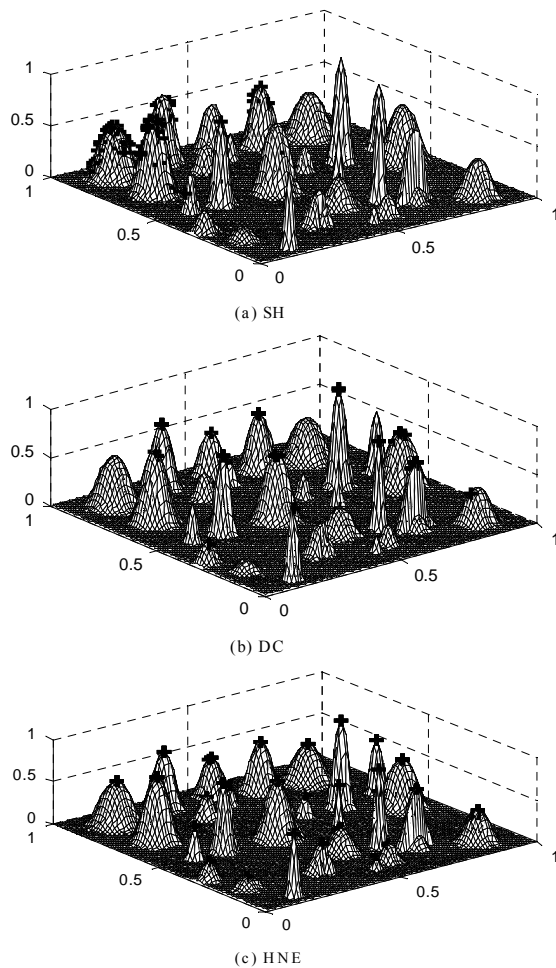


Figure 2 Distributions of the final population of optimizing f_3 the effective classes decrease until the population converges to several optima. Additional experiments show that for those functions have peaks with unequal heights, the population converges ultimately to a single optimum.

For lacking effective hill-climbing mechanism, SH only provides very weak capacity generating effective classes, which can be verified by the trend curve slowly ascending as showed in the lower subfigure in Fig.1. The less values of NEC and the fluctuating curves in equilibrium show that SH can only maintain unstable equilibrium. Thus when the population size is small, SH is likely to lose the effective classes generated, result in unexpected uniform convergence of the population.

Fig.2 plots the distributions of the members in the final population sampled randomly from 100 runs for optimizing f_3 , where the plus signs denote the members. Compared with the subfigure (a) and (b), the plus signs uniformly scattered at the tops of the bells in subfigure (c) vividly demonstrate that HNE has the strongest capacity of parallel exploration and local hill-climbing. For the rest functions, the distributions of the final population members generated by the three test algorithms show the similar characteristics, the plots are not presented in this paper to save pages.

5. CONCLUSIONS

Clustering crowding of HNE decreases replacement errors and genetic drift by extending the area searching similar individual, and analyzing the topological structure of fitness landscape. The combination of deterministic and probabilistic replacement increases the capacity of generating and maintaining effective class in a single population. Real coded representation, Gaussian mutation and PLS speed up the parallel local convergence with lower computation cost. HNE is a class of implicit, self-adaptive niching evolutionary algorithms. The implementation of the algorithm requires no a prior knowledge of the problems and additional control parameters, it can be easily applied to different optimization problems. Compared with typical niching genetic algorithms such as SH and DC, the capacity of generating and maintaining classes, solution precision, convergence speed and the computation cost of HNE are clearly superior to those of SH and DC. The current version of HNE is designed primarily for multimodal function optimization, we are investigating the method analyzing the locality at different representation, involved research will be reported in future publications.

6. REFERENCES

- [1] J.H. Holland, *Adaptation in Nature and Artificial Systems*, 2nd, Cambridge MA: MIT Press, 1992.
- [2] D. E. Goldberg, J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Proceedings of the 2nd International Conference on Genetic Algorithms*, Hillsdale, NJ: Lawrence Erlbaum, 1987, pp.41~49.
- [3] S.W. Mahfoud, *Niching Methods for Genetic Algorithms*, Doctoral Dissertation, Urbana-Champaign: University of Illinois, 1995.
- [4] O.J. Mengshoel, D.E. Goldberg, "Probabilistic Crowding: Deterministic Crowding with Probabilistic Replacement", *Technique Report No.99004*, Urbana-Champaign: University of Illinois, 1999.
- [5] Guo Guanqi, Yu Shouyi, "Evolutionary Parallel Local Search for Function Optimization", *IEEE Transactions on Systems, Man and Cybernetics*, Part B, Vol.33, No.6, 2003, pp.864~876.
- [6] T. Bäck, *Evolutionary Algorithms in Theory and Practice*, New York: Oxford University Press, 1996.
- [7] J.H. Holland, "Building Blocks, Cohort Genetic Algorithms, and Hyperplane-defined Functions", *Evolutionary Computation*, Vol.8, No.4, 2001, pp.373~391.
- [8] H. G. Beyer, *The Theory of Evolutionary Strategies*, Berlin: Springer-Verlag, 2001.
- [9] R. Ursem, "Multinational Evolutionary Algorithms", *Proceedings of 1999 IEEE International Conference on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1999, pp.1633~1640.
- [10] M. Jelasity, T. Dombi, "GAS, A Concept on Modeling Species in Genetic Algorithms", *Artificial Intelligence*, Vol.99, No.1, 1998, pp.1~19.

Constraint-based Concurrency in Java

Rafael Ramirez¹, Juanjo Martinez¹, Andrew E. Santosa²

¹IUA, Pompeu Fabra University
Barcelona, 08003, Spain

Email: rafael@iua.upf.es, juanjo@upf.edu Tel.: +34 93 542 2165

²Department of Computer Science, National University of Singapore
Singapore 117543

Email: andrews@comp.nus.edu.sg

ABSTRACT

Constraint-based synchronization pioneered by (concurrent) logic and concurrent constraint programming is a powerful mechanism for elegantly synchronizing concurrent and distributed computations. They support a declarative model of concurrency that avoids explicitly suspending and resuming computations. This paper describes (1) a model of concurrency based on precedence constraints, (2) its implementation as an extension to the Java programming language, and (3) how model-based verification methods can straightforwardly applied to programs in the resulting language.

Keywords: Concurrency, synchronization, constraints, declarative programming.

1. INTRODUCTION

Concurrent, distributed and parallel systems are becoming increasingly important. However, such systems can be difficult to design, build, and debug. In particular, the synchronization of concurrent and distributed computations imposes serious complications: synchronization code is not neatly encapsulated into a single unit which results in its implementation being scattered throughout the source code. This harms the readability of programs and severely complicates the development and maintenance of concurrent systems. Furthermore, the low-level nature of the synchronization constructs in existent concurrent and distributed programming languages complicates the formal treatment of the concurrency issues of programs which directly affects the possibility of formal verification, synthesis and transformation.

This paper presents a high-level constraint-based model of concurrency and its implementation which conservatively extends Java to allow synchronization via constraint entailment. On the one hand, the model provides a declarative formalism in which concurrency issues are treated as orthogonal to the system base functionality. This provides great advantages in writing, verifying and manipulating concurrent and distributed systems. On the other hand, the implementation of the model results in an extension to Java that makes constraint technology for concurrent programming available in a widely used programming language. Additionally, the implementation contributes techniques for integrating constraint-based synchronization in programming languages based on objects with a predefined concurrency model.

The rest of the paper is organized as follows: Section 2 presents the language used for specifying inter-process

synchronization and communication. Section 3 describes our approach to program verification. Section 4 briefly describes our implementations. Section 5 reports on some related work. In Section 6 we briefly describe our prototype distributed implementation and finally Section 6 summarizes the contributions and indicates some areas of future research.

2. LOGIC PROGRAMS FOR DISTRIBUTED PROGRAMMING

Events and Constraints

Many researchers, e.g. [8, 10], have proposed methods for reasoning about temporal phenomena using partially ordered sets of events. Our approach to concurrent programming is based on the same general idea. The basic idea here is to use a constraint logic program to represent the (usually infinite) set of constraints of interest. The constraints themselves are of the form $X < Y$, read as "X precedes Y" or "the execution time of X is less than the execution time of Y", where X and Y are events, and $<$ is a partial order.

The constraint logic program (CLP) is defined as follows (for a complete description, see [11]). Constants range over events classes E, F... and there is a distinguished (postfixed) functor $+$. Thus the terms of interest, apart from variables, are e , $e+$, $e++...$, f , $f+$, $f++...$. The idea is that e represents the first event in the class E, $e+$ the next event, etc. Thus, for any event X, $X+$ is implicitly preceded by X, i.e. $X < X+$. We denote by $e(+N)$ the N-th event in the class E. Programs facts or *predicate constraints* are of the form $p(t_1, \dots, t_n)$ where p is a user defined predicate symbol and the t_i are ground terms. Program rules or *predicate definitions* are of the form $p(X_1, \dots, X_n) \leftarrow B$ where the X_i are distinct variables and B is a rule body restricted to contain variables in $\{X_1, \dots, X_n\}$. A program is a finite collection of rules and is used to define a family of partial orders over events. Intuitively, this family is obtained by unfolding the rules with facts indefinitely (in general, *reactive* concurrent programs on which we are focusing, do not terminate), and collecting the (ground) *precedence constraints* of the form $e < f$. For example, consider the following program with one rule for p :

$$\begin{aligned} & p(e, f). \\ & p(E, F) \leftarrow E < F, \quad p(E+, F+). \end{aligned}$$

it defines the partial order $e < f$, $e+ < f+$, $e++ < f++$, Multiple rules for a given predicate symbol give rise to different partial orders. We will abbreviate the set of clauses:

$$H \leftarrow C_{s1}, \dots, H \leftarrow C_{sn}$$

by the *disjunction constraint* $H \leftarrow C_{s1}; \dots; C_{sn}$ (disjunction is specified by the usual disjunction operator ';').

Interpreter

The constraint logic program as defined above has a procedural interpretation that allows a correct specification to be executed in the sense that processes run only as permitted by the constraints represented by the program. This procedural interpretation is based on an incremental execution of the program and a *lazy* generation of the corresponding partial orders. Constraints are generated by the CLP only when needed to reason about the execution times of current events. A detailed description of the interpreter can be found in [12]).

The procedural interpretation differentiates our approach with proposals based on point algebra, e.g., [2], in which the focus is on the satisfiability of partial orders. We do not analyze the partial orders, but we provide a procedural interpretation to the partial order specification so we can use it to coordinate concurrent processes.

The coordination mechanism: how processes coordinate

Processes interact with each other by performing simple operations on a shared constraint store. In the distributed case, the constraint store is a shared, network-accessible repository for constraints and objects. Processes can perform operations to *check* if the execution of a particular event is entailed by the constraints in the store, to *write* new objects into the store, or to *read* objects in the store. Synchronization is achieved only by using the \$check\$ operation. The rest of the operations are non-blocking.

- **check(E):** checks whether execution of the current event in class E is entailed w.r.t. the constraints in the store. If it is, **check(E)** has no effect and process execution continues with the next instruction. If execution of the event is not entailed by the constraint store, the process suspends. If an process A executes an operation **check(e)**, and (1) there is a constraint in the constraint store that matches $f < e$, and (2) f has not yet occurred, i.e., **check(f)** by another process B has not yet have any effect on the store, then A suspends. A will be notified when f has indeed been "checked" by B and executed by the store. The store executes an event by recording it as a past event in the history. The execution of f will remove the constraint $f < e$ from the constraint store, thus possibly enabling e to execute and allowing process A to resume execution.
- **write(X):** writes a new object X into the constraint store.
- **take(X):** takes an object matching X from the constraint store. If no object in the store matches X then the process is notified about this.
- **read(X):** reads (make a copy of) an object matching X from the store.

The behavior of the operations described above distinguishes our approach from other coordination models, especially blackboard architectures. Here, the *check* operation is the only way to suspend an process. Thus, process suspension and resumption depends solely on the explicit partial order specify by the constraints. *Check* operations are normally performed at points of interest in the process programs. In the presence of loops and procedure calls in the code of processes, a *check* operation is typically executed several times. Thus, the store keeps track of the *current* event in an event class, e.g. $e, e+, e++$ (e represents the first visit, $e+$ the second, etc.). *check* operations can be seen simply as program comments (i.e. they can be ignored) if only the functional semantics of an process is considered. At the conceptual level, each operation described above is executed atomically in the constraint store.

3. PROGRAM VERIFICATION

We have applied the SMV model checking system [9] to verify properties in our system. We have implemented a prototype which automatically translates an extended Java program into a model M in SMV's description language (a screenshot of this prototype is shown in figure 1). Thus, it suffices to code the property we want to verify using the specification language of SMV resulting in a CTL (computation tree logic) formula p, and run SMV with inputs M and p.

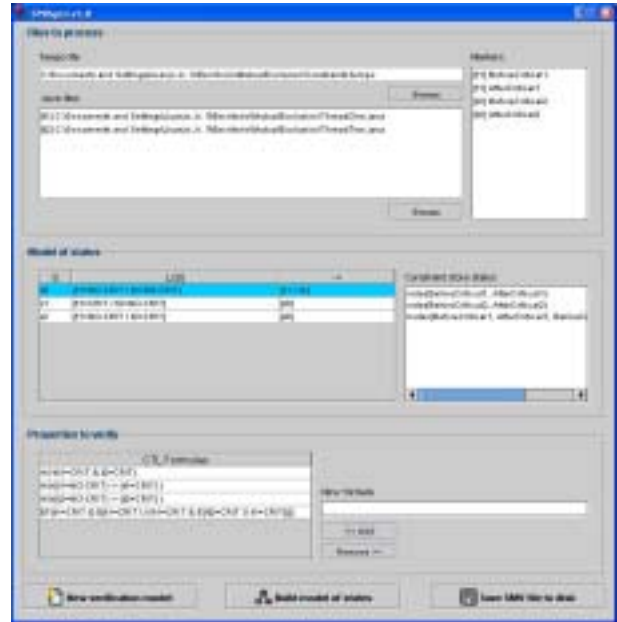


Fig. 1

Event annotations, i.e. *check* operations, naturally divide processes in a (often very small) number of states. Transitions among these states are explicitly determined by the system constraints. Taking into account these constraints it is quite simple to model an event-annotated multi-threaded Java program as a transition system. Each state in the transition system is a collection of possible states of each of the system threads and each transition represents the execution of an enabled event. In order to model a program as a transition system we need to recursively generate all possible reachable states starting with the program initial state. Thus, it suffices to determine the enabled events for each reachable state and generate a new state for each enabled event. It is then straightforward to translate the transition system of the program into a model checker description language and directly apply the model checker to verify program properties. As an illustrative example, consider the mutual exclusion code synchronizing with the constraints:

```
T1 is
while (true) {
    non-critical~state;
    check(a1);
    critical~state;
    check(b1);
}
```

```
T2 is
while (true) {
```



```

check(a2);
non-critical~state;
check(b2);
critical~state;
}

```

synchronizing with the constraints

```

mutex(a1,b1,a2,b2).
mutex(X1,Y1,X2,Y2) ←
  Y1 < X2, mutex(X1+,Y1+,X2,Y2);
  Y2 < X1, mutex(X1,Y1,X2+,Y2+).

```

Abstracting the Java code results in two threads T1 and T2, each of which is in its non-critical state (n) or in its critical state (c). Each individual thread undergoes transitions in the cycle $n \rightarrow c \rightarrow n \rightarrow c \rightarrow \dots$, and the two threads interleave with each other. At any moment in time, the possible transitions for a each thread are determined by the constraints in the store. Starting with the initial state, i.e. both T1 and T2 in their non-critical state, we can obtain automatically both the state transition system, and its translation into code in SMV's description language (Figure 2).

MODULE main

```

VAR
  pr1 : process prc(pr2.st);
  pr2 : process prc(pr1.st);
--safety
SPEC AG!((pr1.st = c) & (pr2.st = c))
--liveness
SPEC AG((pr1.st = n) -> AF (pr1.st = c))
SPEC AG((pr2.st = n) -> AF (pr2.st = c))
--no strict sequencing
SPEC EF(pr1.st = c & E[pr1.st = c U
  (!pr1.st = c & E[! pr2.st = c U pr1.st = c ])))

```

MODULE prc(other-st)

```

VAR
  st : {n, c};
ASSIGN
  init(st) := n;
  next(st) :=
    case
      (st = n) & (other-st = n) : {n, c};
      (st = c)                  : {c, n};
      1                          : st;
    esac;
FAIRNESS running
FAIRNESS !(st = c)
FAIRNESS !(st = n)

```

Fig.2

This code consists of two modules, *main* and *prc*. The module *main* has two instantiations of *prc*. In each of these instantiations, *st* is the status of a thread (saying whether it is in its critical section or not) and *other-st* is the status of the other thread. The *prc* module specifies how the value of *st* can evolve: when it is n, if the other one is n too, it may stay as n or move to c; when it is c, it may move back to n; and when it is n and the other one is c it has to stay in n. In the module *prc*, we restrict to computations paths along which *st* is infinitely often not equal to c and infinitely often not equal to n (this is specified by FAIRNESS !(*st* = c) and FAIRNESS !(*st* = n) respectively).

4. IMPLEMENTATION

Prototype Java implementations for both uniprocessors and distributed systems have been written. Both implementations are articulated in two main components:

- The parser/verifier parses the text file containing the constraints, builds the data structures required by the interpreter (see below), and checks for some possible semantic and syntactic errors in the text file, e.g. infinite loops in predicate definitions, correct number and type of predicate arguments, etc.
- The interpreter is an object which implements the interpreter mentioned in Section 2 (and described in detail in [11]). The interpreter decides whether or not processes suspend upon reaching a check operation during execution. When a process reaches a check operation m, a request is sent to the interpreter to determine whether the current event e associated with m is disabled, i.e. it appears on the right of a precedence constraint $X < e$, or enabled, i.e. otherwise, w.r.t. the system constraints. If e is found to be disabled, the process is blocked until e becomes enabled, otherwise the process proceeds execution at the instruction immediately after m.

The interpreter

The interpreter is based on a number of components which are implemented as class member variables. These components include the constraint store which contains the set of current constraints, the definition store containing the constraint predicate definitions, the current event which records the last executed event in each event class, and the event table which is a hash table storing information about the different events in the system. In the uniprocessor implementation case, a synchronizer object is created with a path to the text file containing the system constraints:

```

Synchronizer sync = new Synchronizer(`C:\\
Constraints.tempo");

```

and a LOG handler may be attached to the interpreter in order to manage and format its messages.

In the distributed implementation case, it is necessary to start a synchronization server which acts a synchronizer of the distributed program. This component is created and started by a separate program:

```

SynchronizationServer sserv = new
  SynchronizationServer(`C:$\\Constraints.tempo", null));
sserv.start();

```

The second parameter of the constructor serves to attach a LOG handler to the synchronization server. Notice that it can be null if the LOG handler is not needed. The synchronization server has its own thread and thus, it is independent of the program that starts it. Some mechanism is required to stop the synchronization service. This mechanism is implemented by the SynchronizationServerShutdown class which has a main method and acts as a generic class to stop the synchronization server. It can be executed either on a shell console as a new program, or by calling its (public static) shutdown method. It is also necessary to create RemoteSynchronizer object on each program of the distributed application. For instance:

```

RemoteSynchronizer sync = new
  RemoteSynchronizer(`127.0.0.1");

```

The constructor's parameter indicates the hosts address or host name where the synchronization server is running. It is

important to add a synchronizer closing session code in each program of the distributed application after each synchronization code and just before the end of the program..

Performance

Our implementation is still in a prototype stage, thus several efficiency issues have still to be addressed. However, the performance of our current implementation is acceptable. The performance issues will be described in a companion paper.

Fairness

Fairness is implicitly guaranteed by our implementation. Every event that becomes enabled will eventually be executed (provided that the program point associated with it is reached). This is implemented by dealing with event execution requests in a first-in-first-out basis. Although fairness is provided as the default, users, however, may intervene by specifying priority events (on how to do this, see [11]). It is therefore possible to specify unfair scheduling.

5. RELATED WORK

The idea of communication using a shared store is not new. There are primitive unstructured mechanisms such as shared memory which provides for a common shared array of words. This is exemplified by systems such as Threadmarks [1]. More related to our work, there are the more structured blackboard architectures well exemplified by coordination languages such as Linda [3]. Linda provides a language-independent model where synchronization and communication is achieved via a shared tuple space. However, the tuple space has no logical reading on its own and it is up to the programmer to give meaning to the tuples on the tuple space. In general, this forces the specification of a system to be low-level and makes impossible any formal treatment for verifying specifications. There are a number of Linda-based programming languages such as Jada [4], Jinni [14], and JavaSpaces [5] which preserve Linda's model of synchronization and communication and thus inherit the disadvantages stated above. Closer to our work are more declarative approaches including concurrent logic programming (e.g. Parlog [6], KL1 [15]) and concurrent constraint programming [13,16]. Although these approaches preserve many of the benefits of the abstract declarative model, such as the logical reading of programs and the use of logical variables, important program properties, namely safety and progress properties, remain implicit.

None of the previous approaches attempt to provide a clear separation of program application functionality and concurrency control. In this respect, i.e. separation of concerns, *aspect-oriented programming* [7] is closer to our work, particularly the work by De Volder and D'Hondt [17]. Their proposal utilizes a full-fledged logic programming language as the aspect language. In order to specify concurrency issues in the aspect language, basic synchronization declarations are provided which increase program readability. Unfortunately, the declarations have no formal foundation. This reduces considerably the declarativeness of the approach since correctness of the program concurrency issues directly depend on the implementation of the declarations.

6. CONCLUSIONS

We have described a high-level model of concurrency based on the idea of a shared constraint store where synchronization is achieved via constraint entailment. In this framework, (distributed) processes are coordinated explicitly by constraints, there is a clear separation of the concurrency aspects in the system from the rest of the code, and model-based verification methods can be straightforwardly applied to our programs. Prototype implementations for both uniprocessors and distributed systems have been written.

Future work. This paper presents work in progress so several important issues are still to be considered. Our implementation is still in a prototype stage, thus several efficiency issues have still to be addressed. In particular, we will focus on how the two key features of incrementality and laziness may be most efficiently achieved. Another important issue is how to deal most efficiently with partial failure. We are also looking into developing a methodology that uses the generative technique for engineering concurrent programs using constraints. Using this technique, programs may be generated using high-level descriptions. The declarative nature of our language particularly fits this approach.

7. REFERENCES

- [1] Amza, C. et al. 1996. TreadMarks: Shared Memory Computing on Networks of Workstations, IEEE Computer, 30(7).
- [2] Broxvall, M. and Jonsson, P. 1999. Towards a complete classification of tractability in point algebras for nonlinear time, Proc. CP'99.
- [3] Carriero, N. and Gelernter, D. 1991. How to write parallel programs: A first course, MIT Press.
- [4] Ciancarini, P. and Rossi, D. 1998. Coordinating Java agents over the WWW. World Wide Web 1(2):87--99, Baltzer.
- [5] Freeman, E., Hupfer, S. and Arnold, K. 1999. JavaSpaces: Principles, and Practice. Addison-Wesley.
- [6] Gregory, S. 1987. Parallel Logic Programming in PARLOG, Addison-Wesley.
- [7] Kiczales, G. et al. 1997. Aspect-oriented programming. In ECOOP~'97-Object-Oriented Programming, Lecture Notes in Computer Science, number 1241, pp. 220--242, Springer-Verlag.
- [8] Kowalski, R.A. and Sergot, M.J. 1986. A logic-based calculus of events. New Generation Computing 4, pp. 67--95.
- [9] McMillan, K.L. 1993. Symbolic Model Checking. Kluwer Academic Publishers.
- [10] Pratt, V. 1986. Modeling concurrency with partial orders. International Journal of Parallel Programming 15, 1, pp. 33--71.
- [11] Ramirez, R. 1996. A logic-based concurrent object-oriented programming language, PhD thesis, Bristol University.
- [12] Ramirez, R., Santosa, A.E., Yap, R. 2000. Concurrent programming made easy, IEEE International Conference on Engineering of Complex Computer Systems, IEEE Press.
- [13] Saraswat, V., and Rinard, M. 1990. Concurrent Constraint Programming. Proceedings of the 7th Annual ACM Symposium on Principles of Programming Languages. pp.232-245.
- [14] Tarau, P. 1999. Jinni: Intelligent mobile agent

programming at the intersection of Java and Prolog. Proceedings of PAAM~99.

- [15] Ueda, K. and Chikayama, T. 1990. Design of the kernel language for the parallel inference machine. *Computer Journal* 33, 6, pp.494-500.
- [16] Van Roy P. et al. 1997. {it Mobile Objects in Distributed Oz}, *ACM Transactions on Programming Languages and Systems*, 9-5.
- [17] De Volder, K. and D'Hondt, T. 1999. Aspect-oriented logic meta programming. In *Meta-Level Architectures and Reflection*, LNCS, 1616, pp. 250--272. Springer-Verlag.



Rafael Ramirez is an Assistant Professor in the Technology Department of the Pompeu Fabra University. He obtained his BSc in Mathematics from the Natonal University of Mexico, and his MSc and PhD in Computer Science from the University of Bristol, UK. Previously, he was a lecturer in the Department of Computer Science in

the National University of Singapore and researcher at INRIA.

Component Based Simulation Environments of Distributed Discrete Event Simulation

Zhang Yaohong, Luo Xueshan, Luo Aiming, Su Wei

College of Humanities and Management, National University of Defense Technology

Changsha, Hunan, China

Email: zhang_yaohong@263.net Tel: +86 (0)731 4573575

ABSTRACT

Component based design is a software design method developing from object-oriented design. It uses hierarchical, modular ideas to analyze and design systems. It improves the reusability of software, decreases the cost of system developments. This paper applies the ideas of component based software design and distributed discrete event simulation (DDES), puts forward simulation component model standards and distributed simulation method, and develops the simulation environment. This method allows the users to reuse existing models and to build simulation by assembling basic models. It is fit for modeling and simulating large and complex systems in especial domain such as communication network, and supports the reuse of models effectively. The simulation environment has good flexibility and expansibility.

Keywords: Component DDES Simulation Environment

1. INTRODUCTION

In software design domain, the component is the software module that can be reused. The component has several interfaces, and each interface represents one of the attributes or methods of the component. The other components or application systems can do some operations by setting these attributes and employing methods. Now there are three standards that we can choose:

COM/DCOM: It is based on the Microsoft COM (Component Object Model). Its code can be written in any windows program language, for instance, VB, Delphi, C, C++, and can run in any windows applications.

CORBA: Its component can be written in C, C++, Java, Small Talk or Ada. Which language you choose depends on the implement vision of CORBA. The component of CORBA can run in Windows, Unix and many other applications. Its standard is IIOP (Internet Inter-ORB Protocol).

JavaBeans: It is based on the definition of JavaSoft JavaBeans. Its component is written in Java, and can run in any Java applications. It uses various communication mechanisms, including Java RMI (Remote Method Invocation) and IIOP, such that it can be used in many applications.

Compared with the traditional software design method, the component based design regards the development of software as an assembling process. Under the direction of software component the developers find the components that can be reused, or develop more new components, and then develop system with them. While the systems being simulated become more and more complex, the workload of developing simulation applications increase greatly. If we start from

scratch to develop a new simulation system, the efficiency will be very low and the reliability of the simulation system can't be ensured. With the component based software design, we can use many existing simulation models to develop a huge simulation system, which is an effective way to deal with the difficulty in complicated system simulation. The technology had been noticed by more and more people (Miller 1998, Buss 2000). And some simulation environments can be available, for example, J.A. Miller's Jsim [1][2].

The remainder of this paper is organized as follows. Section 2 introduces component-base design method. Section 3 states the structure of simulation model component and the realization of information exchange among components. Section 4 gives a brief overview of our distributed simulation synchronization method. Section 5 introduces our simulation environment CBMSE (Component-Based Modeling and Simulation Environment) that is based on Microsoft COM standards.

2. COMPONENT BASED DESIGN

Component based design is a design method mainly by assembling components. At first you search for the available general components of your domain, then develop new components to meet the system's special requirements, and finally assemble the general components and the special components to a whole system, as shown in figure 1.

- **Simulation domain analyze**
Collect the interrelated knowledge and experience of simulation domain, establish the framework and requirements of components.
- **Simulation components develop**
Develop the simulation model components according to the general component standard.
- **Simulation components test**
Test functions of the simulation components.
- **Simulation components submission**
Submit the components to the component database.
- **Simulation system develop**
According to the problem domain and user's requirements of special simulation application, develop new components and integrate them with available components, build simulation system at last.

By this means building simulation system need not start from scratch, the developer can shorten the developing cycle and improve the developing quality. However there should be enough components to use, furthermore component standard should be established to ensure reusability.

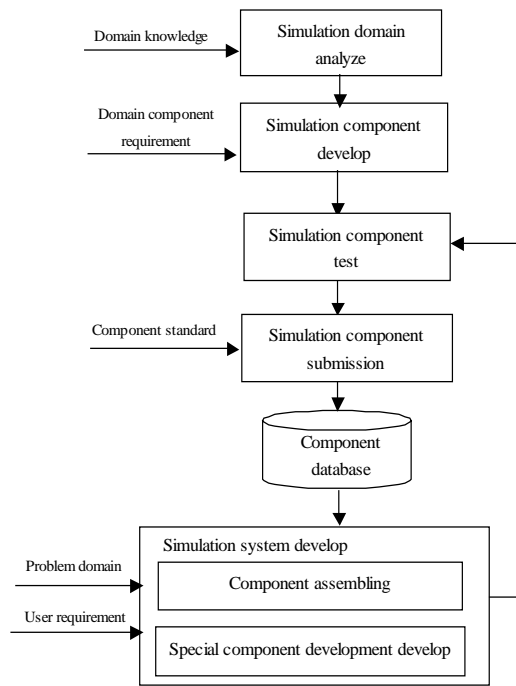


Figure 1. Component based design method

3. SIMULATION MODEL COMPONENT

Petri net model, Queue model and other discrete event model can realize the discrete event simulation models. The automatic model of DEVS Theory can describe these models [3][4]. Zeigler's formal representation of a model can be defined as follow.

$$M = \langle X, S, s_0, Y, \text{int}, \text{ext}, \text{ta} \rangle \quad (1)$$

Where

X : the input event set

s_0 : the initial state set

S : the state set

Y : the output event set

$\text{int}: S \times \{i\}$ S : the inside transfer function, i represents the inside event of the next time

$\text{ext}: Q \times X$ S : the outside transfer function, Q represents state set, such that: $\{(s, e) | s \in S \text{ and } 0 \leq e \leq \text{ta}(s)\}$

$: S \times \{i\}$ Y : output function

$\text{ta}: S \rightarrow \mathbb{R}^+_{\infty}$: time schedule function

Considering the characteristics of discrete event simulation model, we can transform the automatic model to a software component model. The basic structure can be represented as

$$M = \langle P, M, E, \text{Port} \rangle \quad (2)$$

where P is attribute set, M is method set, E is event set and Port is port set, which include in port sets and out port sets, that is:

Attribute is used to describe the properties of the model. It includes version number, run environment, design environment, function description, component name and so on. One can find components by searching attributes of them.

Method is used to describe the actions of the object. The methods that can be accessed from outside are called interface, which contain general interfaces and private interfaces. Component must have all of the general interfaces, for example:

```
void SimInit (double fSimClock)
Components initialize simulation process.
void SimStart (double fSimClock)
Components start simulation process.
void SimStop (double fSimClock)
Components stop simulation process.
```

Simulation environment controls and manages components by general interfaces.

Event is the action that causes the changes of model states. Port is the interface of information communication between model and simulation environment. There are two kinds of ports: input ports and output ports. Models send message to environment via input ports, and receive message via output ports. There is no direct communication channel among models. Simulation environment realizes information exchange among models by dispatching messages.

We assign each port to a message type. If the message type of the output port of a model is same with that of the input port of another model, there is a connection between the two ports of the two models. For example, if the simulation environment receives a message type "comm", it will look up all the models in the environment, find the one whose input port message type is also "comm", and send the message to it, as shown in Figure 2.

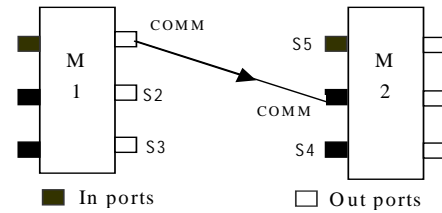


Figure 2. Model information exchange

In this way, when models send messages, it doesn't mind which model to receive and how to send messages. Simulation environment will decide the model, which should receive message and dispatch message wherever the model runs. All of models' properties are private, which makes models have better reusability.

The basic models are assembled into coupled models by connecting the ports. The output ports and input ports of the coupled model corresponds the output ports and input ports of the submodels, as shown in Figure 3.

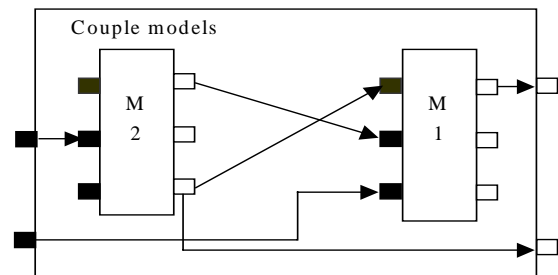


Figure 3. Model coupling

In our system, we use simulation engine to control the running of the models. Every model must be in accordance with the

simulation engine before its clock can move forward. While the model is running, it must report the local clock to the simulation engine. In the simulation, the event list is used to save the messages produced when the models run. The messages in the event list wait for being scheduled by the simulation engine. The simulation engine looks for the minimum next event time of all the models, and uses it as the next time. The flow chart of the simulation arithmetic of simulation engine can be shown as Figure 4, the current time of the simulation environment is t and the next event time is tN .

The synchronization arithmetic of simulation engine is very simple, and doesn't bring deadlock. It also can manage and control the whole simulation task; such as start, pause, resume and stop.

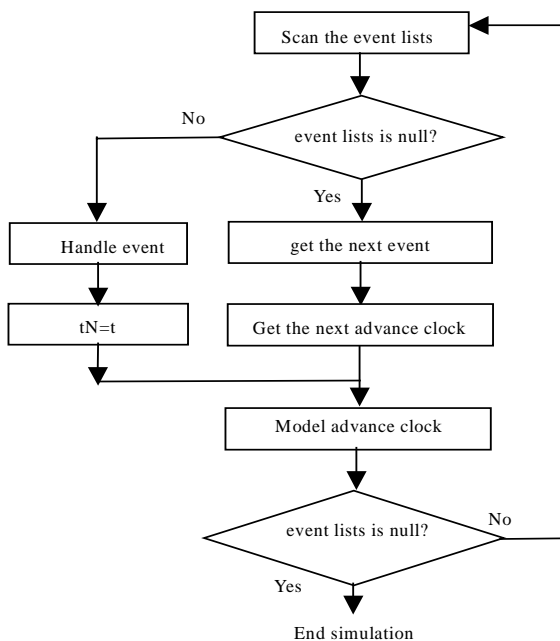


Figure 4. Simulation synchronization method

4. SIMULATION ENVIRONMENT

Based on the concept and methods as described above, we have developed simulation environment CBMSE, such as figure 5.

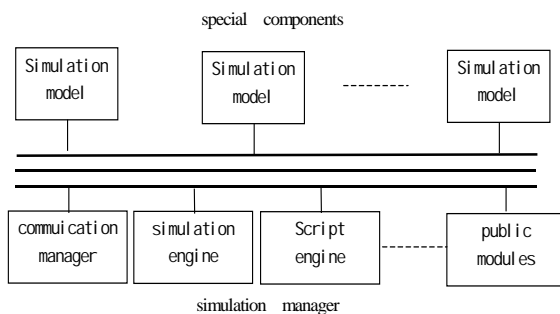


Figure 5. Simulation environment framework

Simulation environment has a frameworks based on

simulation bus. Simulation bus is employed as a communication infrastructure which supports distributed simulation. Models and simulation manager exchange information by simulation bus like computer bus. Simulation manager consists of simulation engine, script engine and other public modules designed with component technology. Simulation models run by using interfaces of simulation manager, and simulation manager controls models by using interfaces of models in the same way. These interfaces are general in all simulation domains, user only focus on information exchange relation in models, which is different in special applications.

Figure 6 describes main modules and tools of CBMSE, for example:

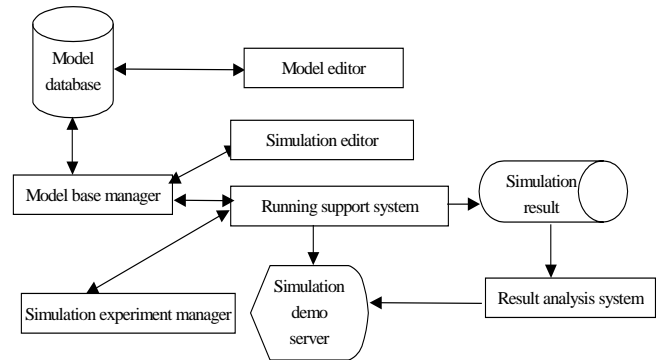


Figure 6. Simulation environment

The model editor is used to develop the model of a simulation system. Its' developing language can be VC, VB, etc. Models must be designed according to Microsoft COM standards.

The simulation editor is a visual integrated edit environment, which integrates the graph and text editor. Users can select basic model components from the model component database, describe information exchange relation between models, assemble them and initialize the model.

The simulation model database saves the basic model components. The model database manager takes charge of the maintenance of component database, including deleting, adding, copying the components.

The simulation experiment manager can configure and manage the simulation environment. During the simulation running, it is the operation interface between the users and the simulation system. It can configure the simulation experiment, receive the user's command, and control the simulation process.

The running support system realizes the communication and information share among the models. It supplies necessary supports for simulation, realizes the communication among models, manager model objects, including creating, deleting and moving the objects. It hides the details for the higher model developer such as communication in a heterogeneous environment; users only need to focus on the details of special application.

Simulation demo server demonstrates event, state change, and model action in simulation process.

Result analysis system collects data and information, provides simulation analysis result.

xiaochenxing@163.com

5. CONCLUSIONS

We have used this simulation environment in many applications, such as the armored division battle simulation. The simulation environment can realize modeling and simulation of complex system. The model has better reusability than before. It is good for some special simulation domains.

6. REFERENCES

- [1] A.H.Buss, "Component-Based Simulation", Proceedings of the 2000 winter simulation conference
- [2] Miller J A, "Component-based Simulation Environments: Jsim As a Case Study Using Java Beans", Proceedings of the 1998 Winter Simulation Conference, 373-381.
- [3] B.P.Zeigler, "Hierarchical, Modular Discrete-Event Modeling in a Object-Oriented Environment", simulation, 1987, 5: 219-230.
- [4] B.P.Zeigler, Theory of Modeling and Simulation, 2nd Edition, Academic Press, 2000
- [5] Chandy K.M and Misra J, "Distributed simulation: A Case Study in Design and Verification of distributed programs", IEEE Trans. on Software Eng. 1979, SE-5 (5): 440-452
- [6] Fujimoto R M, "Time warp on a shared memory multiprocessor", Intl. Conf. on Parallel Processing, 1989, III: 242-249.
- [7] Misra J, "Distributed Discrete-Event Simulation", 1986 Computer Surveys, 18:39-65



Zhang Yaohong is an Associate Professor at College of Humanities and Management in National University of Defense Technology, he received PhD in National University of Defense Technology in 2000. His research interests include discrete event simulation, system analysis and design technology, with applications to performance analysis of communication networks. His e-mail address is zhang_yaohong@263.net

Luo Xueshan is a Professor at College of Humanities and Management in National University of Defense Technology; he received PhD in National University of Defense Technology in 1992. His research interests include discrete event simulation, information system engineering. His e-mail address is xsluo@public.cs.hn.cn

Luo Aiming received the B.S, M.S degrees in electrical engineering from National University of Defense Technology in 1991 and 1994 respectively. In 1994, she joined the College of Management, NUDT, where she is an Associate Professor. Her current research interests include C3I theory, system analysis and application simulation. Her e-mail address is luoaaiminnudt@sina.com

Su Wei received the B.S, M.S degrees from National University of Defense Technology in 2000 and 2002 respectively. Now he is studying the PhD of industry. His research interests include discrete event simulation, information system engineering. His e-mail address is

An Approach of Component Tailoring Based-on Parameterized Contracts *

Fei Yui-Kui, Wang Zhi-Jian

College of Computer and Information Engineering, Haohai University

Nanjing 210098, China

Email: feiyukui1@sina.com Tel: +86 (0) 025-83789902

ABSTRACT

Component tailoring is a common operation of building system out of components because of varieties of context of reusing components. As a generalization of interoperability check between components, parameterized contracts can be applied to component tailoring. Having discussed the principle of the component behavior contracts & parameterized contracts, this paper gives a way of how to apply parameterized contracts to component tailoring operation in component composition environment.

Keywords: component tailoring, parameterized contract, component composition, precondition, post-condition, provides-interface, requires-interface

1. INTRODUCTION

The aim of software engineering in general is to support the efficient development of high-quality software products in such a way, that success is repeatable. The reuse of approved work is one technique to reach this goal. Several reuse techniques have been proposed and employed. None of them has been satisfying, mainly because units of reuse have not been easily adaptable to several specific application contexts (or only with an enormous effort). The development of complex applications can only be accomplished by composing, and thus reusing, approved parts. This composition requires that reuse units must only have explicit external dependencies. To reuse components in assembling an application, they have to be adapted to the specific requirements.

Component is usually made by the third-party. Although makes component himself at times, the user of components usually builds application systems out of components made by others. That brings out a problem: A component rarely fits directly into a new reuse context. For a component developer it is hard to foresee all possible reuse contexts. Hence, it is also hard for a developer to provide components with reasonable configuration options to fit into future reuse contexts. While making use of components, the component user has to make further adaptation operations on them again. Only so that can the components be applied in application system. The process of that is referred as component tailoring. Component tailoring is proceeded at three level [1]: Tailoring by changing parameters of single components; Tailoring by changing the composition of components; Tailoring by changing or extending the implementation of components.

There is lot of methods of Tailoring by changing parameters of single components. For example, wrapper, active interface

binary component adaptation etc. What the common principle is changing parameters of component interface to match the application requirement by means of add-one. The third requires changing component itself. This paper mainly deals with the second method, it will give a focus on how to tailor component by means of component-parameterized contracts.

The structure of this paper is as follows. In the following section component contracts is mentioned. There is a consensus on parameterized contract. Component tailoring based on parameterized contracts is discussed in section 3. After the presentation of related work (in section 4), we conclude with a summary in the last section.

2. COMPONENT CONTRACTS

Contract is a temporary architecture. It composes different components by specifying communication paths and workflows. The components of contract is referred as roles. Every role must obey to some constraints. More precisely, Beugnard et al [3] categorize contracts in four levels:

- 1) Syntactic contracts, that is signatures of the data types.
- 2) Behavioral contracts, that is some semantic description of data types,
- 3) Synchronization contracts, which deal with concurrency issues.
- 4) Quality of Service (QoS) contracts, which encompass all non-functional requirements and guarantees.

In this paper, we only focus on the behavioral contracts. The first level of contract corresponds to type signatures, and type checking is usually performed statically. The synchronization aspects of contracts still need to be studied in a general enough component framework, as concurrency issues are often reduced to the means of communication of a given connector of a component.

Behavior Contracts

According to Meyer [2], a contract is a collection of assertions that describe precisely what each feature of the component does and does not do. The key assertions in the design by contract technique are of three types: invariants, preconditions, and post-conditions.

An invariant is a constraint attached to type that must be held true for all instances of the type whenever an operation is not being performed on the instance. We can attach invariants to an interface to specify properties of the component objects that implement the interface. For example, an invariant might state that the value of some attribute is always greater than zero.

Preconditions and post-conditions are assertions attached to an operation of a type. A precondition expresses requirements that any call of the operation must satisfy if it is to be correct. A post-condition expresses properties that are ensured in

* Supported by the National Grand Fundamental Research 973 Program of China under Grant No. 2002CB312002, the National High Technology Development 863 Program of China under Grant No. 2001AA113170.

return by the execution of the call. In our approach the precondition gives the contractual requirements on the client (that is, caller) of the interface and the post-condition gives the corresponding contractual requirement on the supplier of the operation, the component object that implements the interface. For example, an operation to delete a record from a collection might have a precondition requiring that a record with that key exists and a post-condition requiring that it no longer be an element of the collection.

Assertions are logical expressions about the entities in an interface's information model. They give component developers a precise description of the behavior of a component implementing the interface. A component can only be considered as implementing an interface if all instances of the component satisfy all the assertions in the interface's contract. (If a component implements an extended form of an interface, then it also implements the base interface.) Contracts are important in providing support for reuse of components. If a set of interfaces F of a component is needed in an application, then any component that implements all interfaces in the set F may be used, that is, may be plugged into the application.

Assertions can also help improve the reliability of a component. They are checked at runtime to help test and debug the implementation. A precondition violation indicates a bug in the client. The client did not observe the conditions imposed on correct calls. A post-condition or invariant violation is a bug in the supplier. The supplier failed to deliver on its promises.

Parameters Contracts

A component rarely fits directly into a new reuse context. For a component developer it is hard to foresee all possible reuse contexts. Hence, it is also hard for a developer to provide components with reasonable configuration options to fit into future reuse contexts. This means, that in practice one single pre- and post-condition of a component will not be sufficient, because of the following common cases:

- 1) The precondition of a component is not satisfied by a specific environment while the component itself would be able to provide a meaningful subset of its functionality.
- 2) A weaker post-condition of a component is sufficient in a specific reuse context. For example, the component user might not require all functions. Hence the component will itself require less functionality via its requires interfaces and hence weaken its component precondition.

To model this we need some sort of adaptive pre- and post-conditions. We call these parameterized contracts [4]. In case 1) a parameterized contract computes the post-condition dependent upon the strongest precondition guaranteed by a specific reuse context. Hence the post-condition is parameterized with the precondition. In case 2) the parameterized contract computes the precondition dependent upon the post-condition (which acts as a parameter of the precondition). For components this means, that provides and requires-interfaces are not fixed but are computed to some extent taking into account the reuse context. Hence, in contrast to classical contracts, one can say:

Parameterized contracts link the provides- and require interface(s) of the same component. They range over many

possible actual contracts (i.e., ultimately interfaces). More simply, we can define two kinds of parameterized contracts.

- 1) Provides-parameterized contracts map the provides-interface to a requires-interface.
- 2) Requires-parameterized contracts map the requires-interface to a provides-interface.

Technically spoken, parameterized contracts are a mapping, which is bundled with the component and computes the interfaces of the components on demand. The requires-parameterized contract takes as arguments the requires-interface of the component and the provides-interface of the environment. Hence, parameterized contracts are isomorphic mapping between the domain of preconditions and the domain of post-conditions. The intersection of a component requires-interface and the environment provides-interface describes the functionality, which is required by the component and provided by the environment. Out of that information, the requires-parameterized contract computes the new provides-interface of the component. Analogously, a provides-interface computes the new requires-interface out of the provides interface of the component and the requires-interfaces of its clients.

Like classical contracts, parameterized contracts depend on the actual interface model and should be statically computable. In any case, the software developers do not have to foresee possible reuse contexts but has to provide a bi-directional mapping between provides- and requires-interfaces. For simple interface lists (signatures a la CORBA IDL say), this means, that for each provided service, a list of required external services must be provided by the component developer. When computing the actual provides interface, a service would only be included, if all its required services are provided by the component's environment. If interfaces also describe component protocols, one has to specify a mapping from the provides- interface to the requires interface protocol which also identifies the order in which requires services are invoked.

3. THE IMPLEMENTATION OF COMPONENT TAILORING UNDER PARAMETERIZED CONTRACTS

Applying parameterized contracts to software components composition means that the interfaces of the component are recomputed dynamically. The code has not to be manipulated. In component composition phase, there are two cases occurring: on the one hand, the composition framework state the functionality a component has to fulfill and finds candidate components to realize; on the other hand, integrating components into exiting composition framework to reconfigure or enhance system's functions. In the following, we discuss component tailoring under these cases respectively.

Building System by Component Tailoring

System building is carried out under composition framework. Composition framework contains a series of rules to realize application. In every step, the composition framework states the functionality a component has to fulfill. Then she finds several candidate components in a repository, which deliver at least the required functionality. For all theses candidate components one can compute the functionality they really

need in this context via their provides-parameterized contracts. Some cases may occur.

If the functionality the candidate component really need from environment is satisfied then it is integrated into framework. If the functionality the candidate component really need from environment is not satisfied completely then re-computing the required parameterized contract of the component to get the provide interface, and select other components which can provide the remain functionality at the given preconditions. Composing these components at parallel way to implement the functionality .An other way of implementing the functionality is composing these components in sequence.

If the function provide by the candidate component is redundant then re-computing the provides-parameterized contract of the candidate component to restrict the provide interface. It is based on the observation that users actually only

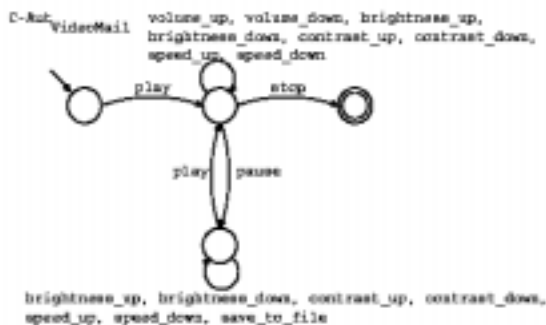


Fig 1 Example: C-Aut of a videoMail component (unrestricted)

use a subset of a component's functionality. So a restricted functionality is often sufficient for the user. More important than a full functionality is that the user has not to provide many other infra-structural resources (e.g., libraries, other components, but also system updates, etc.) which are only necessary to support the part of a component's functionality, the user actually does not need.

As an example, regard a multi-media video-mail component. This mail not only contains the video itself, but also offers functionality to present the video. This design is useful, if you want to abstract away from specific video file formats or if you want to handle different media (like text, sound, and video) in the same manner. The videoMail component makes use of two other (system-specific) component: videoPlayer and soundPlayer. The functionality offered by video-player contains in its provides-interface the methods start, stop, pause, volumeUp, volumeDown, and certain methods to adjust the picture, like brightnessUp, brightnessDown, etc (Fig1). In case videoMail arrives on a system without sound support (e.g. due to hardware reasons) a require-parameterized contract computes a restricted provides-interface without the methods volumeUp and volumeDown (Fig2).

Enhanced Or Reconfigured System Functionality By Component Tailoring

Imagine an existing system should be reconfigured with a new component, or an existing system architecture should be enhanced by an existing component .One question in this situation is which functionality the new component will

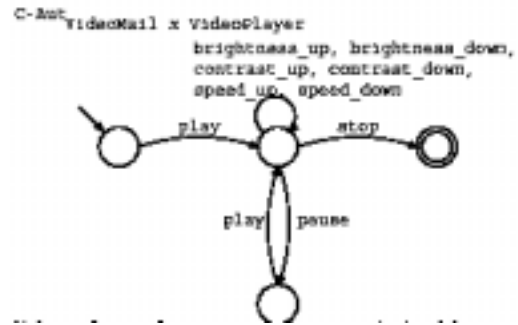


Fig 2 Example: C-Aut of a videoMail component (restricted)

deliver without changing the environment of the component. What we can do is to compute the requires-parameterized contract of the component to obtain the provide function of the component.

4. RELATED WORK

Meyer introduced the idea of "Design by Contract" (DbC) in [2] to increase the reliability and correctness of object-oriented (OO) software by introducing a set of principles to deal with software errors systematically.

Rausch [5] introduces requirements/assurances contracts. Each component is described individually indicating what it requires from its environment, and what the component assures it will provide, given its requirements are met. To ensure contract conformance, the system designer has to "prove" the correctness of the syntax and behavior of each member of the contract. Rausch proposes to start with the conjunction of all predicates assured in the contract and ensure it ends with all required predicates. If a component changes, the contract can be re-checked to ensure all the required predicates can be reached. Requirements/assurances contracts provide a way to ensure designers are aware of the consequences of component changes.

Geise proposes to specify components using Object Coordination Nets (OCoNs) [6], a variant of Petri Nets [BRR87], for specifying the behavior and synchronization aspects of object-oriented systems. According to Geise, contracts should not only include a description of protocols and coordinating sequences but a functional specification that details the pre- and post-conditions, and non-functional properties.

Hummes and Merialdo present an approach for component-based tailoring by extension [7] based on the extensibility pattern. This pattern allows one to make some changes at run-time, provided that these changes conform to an interface. Some existing behavior can be replaced or some new behavior can be added at certain points called "hot spots". These hot spots must be discovered in the design phase and then implemented according to the pattern.

Teege proposes yet another approach to achieve component-based tailoring, particularly tailoring by composition [8]. This approach concentrates on the use of features and parts. A feature represents a system component (part) whose properties or functions can be integrated with other components by simply selecting its presence in a set.

This characteristic is regarded as pure integration.

Syri uses another pattern, the mediator pattern, to provide component-based tailorability [9]. There are three basic object types according to this pattern, viz., target objects, enablers and mediators. Target objects provide the intended cooperation support. Enabler objects encapsulate functionality for basic cooperation support, such as communication, coordination and sharing, and provide this functionality to target objects. A mediator object mediates the interactions between a target object and its associated enablers.

5. CONCLUSION

This paper discussed contractual usage of software component. We present requires interfaces as precondition of components and provides interfaces as post-conditions. Parameterized contracts then link provide and require interfaces of the same component. They are motivated by the necessity of computing functional and extra-functional component properties dependent upon deployment context.

As a generalization of interoperability check between components, parameterized contracts can be applied to component tailoring, especially to tailoring of composition way. The base of parameterized contracts lies in the observation that in most practical cases the provides-interface and the requires-interface of a component are not isolated: A component will offer less functionality if its environment offers not all functionality the component requires. And, a component will require less functionality, if not all offered functionality of the component is to be used by its clients. Hence, it enhances the component's reusability to compute the component's provides-interface out of its required-interface and vice versa.

6. REFERENCES

- [1] Stiemerling, O., Cremers, and A.B: Tailorable Component Architectures for CSCW-Systems. Proceedings of the 6th Euromicro Workshop on Parallel and Distributed Programming, Jan 21-24, Madrid, Spain, IEEE Press, pp. 302-308, 1998.
- [2] B. Meyer. "Applying Design by Contract," Computer, IEEE, October 1992, pages 40- 51.
- [3] Beugnard, A., J.-M. Jézéquel, N. Plouzeau and D. Watkins. Making components contract aware. In IEEE Software, pages 38-45, June 1999.
- [4] R. H. Reussner. The use of parameterised contracts for architecting systems with software components. In W. Weck, J. Bosch, and C. Szyperski, editors, Proceedings of the Sixth International Workshop on Component-Oriented Programming (WCOP'01), June 2001.
- [5] A. Rausch. Software evolution in componentware using requirements/assurances contracts. In Proceedings, International Conference on Software Engineering (ICSE 22), pages 147-156. ACM, June 2000.
- [6] H. Geise, J. Graf, and G. Wirtz. Closing the gap between object-oriented modeling of structure and behavior. In 2nd International Conference on the Unified Modeling Language, Fort Collins, CO, October 1999.

- [7] Hummes, J. and Merialdo, B.: Design of Extensible Component-Based Groupware. In Computer Supported Cooperative Work: The Journal of Collaborative Computing, 9 (1), pp. 53-74, 2000.
- [8] Teege, G.: Users as Composers: Parts and Features as a Basis for Tailorability in CSCW Systems. In Computer Supported Cooperative Work: The Journal of Collaborative Computing, 9 (1), pp. 101-122, 2000.
- [9] Syri, A.: Tailoring Cooperation Support through Mediators. In Proceedings of the Fifth European Conference on Computer Supported Cooperative Work (ECSCW '97), pp. 157-172, 1997.



Fei Yu-Kui, male, born in 1964. He is now a Ph.D. candidate of college of computer & information engineering, Haohai University. His research interests include software reuse, software component, and formal specification technology.

On a Smart Software for Cement-Meal Batching Computation*

Jiang Hongzhou Li Juanjuan

School of Materials, Wuhan University of Technology

Wuhan, Hubei Province, 430070, P. R. China

Email: jianghongzhou@sina.com Tel: 001-86-27-87372753

ABSTRACT

Like in other engineering fields, computer technology and its corresponding internet technology as well as intranet technology also finds its application in cement manufacture industry. In this paper, the authors introduce a kind of smart software, as an important knot of above grid technology, used for cement-meal batching computation in cement manufacture industry. The focus of this academic paper is on the principle and essences of developing the software. Feedback opinions from some users of this software have proved this software for cement-meal batching computation is a kind of smart software with certain expertise effects, which is useful in cement manufacture industry.

Keywords: Computer; Software; Cement; Meal; Batching; Computation

Nowadays, the application of computers as well as computers' web technology or computers' grid technology can be traced in almost every science and engineering fields. As we know, cement is a kind of very important building material and structure material. Cement manufacture industry is a heavy industry field, which plays a vital role in modern society. This paper introduces a kind of smart software for cement-meal batching computation in cement manufacture industry.

In fact, computer technology found its application in cement industry in 1980's [1][2][3]. However, conventional computer programs for cement-meal batching computation, or called forward controlling, is based on more simple mathematical computation.

As time has been progressing as well as science & technology, especially computers' web technology, has been developing, modern cement plants' production controlling system has also entered the era of intranet controlling. This modern intranet controlling system connects every producing process, every production workshop, and every relative department in order to optimize the cement production and obtain higher over-all producing efficiency. Information and all relative data obtained are shared by all computers in the system. This means a package of smart and updated computer software used for cement producing process, compatible to other software in the system, play its indispensable role in the intranet controlling system for cement producing process. Besides, modern technical design processes for cement plant also require smarter computer software in order to optimize technical design. Fig.1 (see next page) is the brief diagram for a computer intranet controlling system used in cement plants. Fig. 2 (also see next page) is the brief flow diagram of design process for cement plants. From the two figures, we can see

that cement-meal batching computation is an important knot in above intranet computer controlling system for cement plants and is also a key in modern technical design processes. Therefore, the computation software for cement-meal batching computation is required to satisfy the optimization of more accurate operation and more reasonable design [2]. The smart software, introduced in this paper, is one of these new kinds of software. Some advantages of the smart software for batching computation in cement industry, developing by us, lie in as follows. • It can respond to errors automatically in order to let users select or adjust errors conveniently according to their requirements. • It can prevent input mistakes or sampling faults to the most extent. • If the raw materials given can not form the cement-meal that the users need, it can give corresponding suggestions to let users know how to change some raw materials or add other appropriate corrective materials.

1. HOW TO RESPOND TO ERRORS AUTOMATICALLY

In order to get a kind of software that can respond to errors automatically, we selected the following mathematical model [1] [3].

As we know, cement is an artificial material, which is made from several raw materials. The raw materials are firstly mixed, according to their respective proportion, and ground into a powder mixture that is called cement-meal, or briefly called meal. The meal is then fired and clinkered into a new kind material that is called cement-clinker or called clinker briefly. After clinker is ground along with some additives, such as gypsum and slag etc., it will become a kind of active powder material, which is called cement. The batching computation in cement industry is mainly to determine the proper proportion of respective raw materials based on three modules of clinker. In China mainland, the three modules are KH (KH denotes the lime saturation factor), SM (SM denotes silica module) and IM (IM denotes alumina module or iron module). By and large, cement meal is formed with three kinds of raw materials, as well as ash of fuel coal burned. Based on the mass conservation principle as well as definition of above three modules, the following equations can be derived.

$$A_{11}X_1 + A_{12}X_2 + A_{13}X_3 = B_1 \quad \text{Eq. (1)}$$

$$A_{21}X_1 + A_{22}X_2 + A_{23}X_3 = B_2 \quad \text{Eq. (2)}$$

$$A_{31}X_1 + A_{32}X_2 + A_{33}X_3 = B_3 \quad \text{Eq. (3)}$$

$$A_{41}X_1 + A_{42}X_2 + A_{43}X_3 = B_4 \quad \text{Eq. (4)}$$

Here, X_i denotes the percentage of the No.i raw material in cement-meal on burned base.

In above equations,

$$A_{1i} = 2.8KH \cdot S_i + 1.65A_i + 0.35F_i - C_i$$

$$A_{2i} = SM \cdot (A_i + F_i) - S_i$$

* This paper is sponsored by the foundation fund of Key Laboratory for Silicate Materials Science and Engineering (Wuhan University of Technology), Ministry of Education, P. R. China

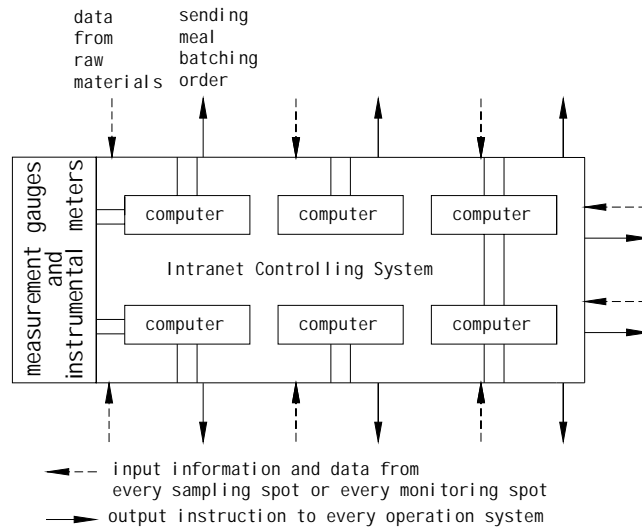


Fig. 1 Brief Diagram of Intranet Controlling System Used in Cement Plants

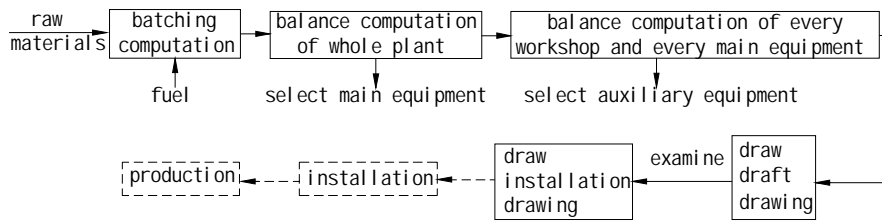


Fig. 2 Brief Flow Diagram of Design Process for a Cement Plant

$$\begin{aligned}
 A_{3i} &= IM \cdot F_i - A_i \\
 A_{4i} &= 1 \\
 B_1 &= -G_A \cdot (2.8KH \cdot S_A + 1.65A_A + 0.35F_A - C_A) \\
 B_2 &= -G_A \cdot [SM \cdot (A_A + F_A) - S_A] \\
 B_3 &= -G_A \cdot (IM \cdot F_A - A_A) \\
 B_4 &= 100 - G_A
 \end{aligned}$$

where S_i , A_i , F_i and C_i denote the percentage of silica (silicon dioxide), alumina (aluminium sesquioxide), ferric oxide (iron sesquioxide) and lime (calcium oxide) in No. i raw material on burned base, respectively, and S_A , A_A , F_A and C_A denote the percentage of silica (silicon dioxide), alumina (aluminium sesquioxide), ferric oxide (iron sesquioxide) and lime (calcium oxide) in ash of fuel coal burned, respectively, and G_A denotes the percentage of ash of fuel coal burned in cement-meal and KH , SM and IM is the target value of the three modules KH , SM and IM , respectively.

It is obvious that the equations (1) – (4) (four equations) have three unknown quantities X_1 , X_2 and X_3 . So, the equations (1) – (4) do not have a unique solution. In other words, any solution of the equations (1) – (4) must have errors for above three modulus. That means, when three

raw materials are used to form the cement-meal, it is inevitable that there are errors between the computation values of the three modules KH , SM , IM and their corresponding target values of above three modules.

In order to get the solution of the equations (1) – (4), we can carry out the following mathematical transformations.

Firstly, the equations (1) – (4) can be represented as the matrix equation:

$$A \cdot X = B \quad \text{Eq. (5)}$$

in which A is the matrix (A_{ij}) ($i = 1 - 3$, $j = 1 - 4$), X is the matrix (X_i) ($i = 1 - 3$), and B is the matrix (B_j) ($j = 1 - 4$).

Then, introduce a diagonal weight coefficient matrix $R = (R_{ij})$ ($i = 1 - 4$, $j = 1 - 4$), in which $R_{11} = R_1$, $R_{22} = R_2$, $R_{33} = R_3$, $R_{44} = R_4$, if sub index $i \neq j$, then $R_{ij} = 0$. And transform the matrix equation (5) into the following form.

$$R \cdot A \cdot X = R \cdot B \quad \text{Eq. (6)}$$

From the matrix equation (6), it is easy to understand that R_1 , R_2 , R_3 and R_4 are actually the weights of the equation (1), the equation (2), the equation (3) and the equation (4),

respectively. Through further analysis, we can know that R_1 , R_2 , R_3 are the weights of the modules KH, SM and IM, respectively. From the matrix equation (6), we can also get the following matrix equation through further mathematical transformation.

$$A^T \cdot R \cdot A \cdot X = A^T \cdot R \cdot B \quad \text{Eq. (7)}$$

In the matrix equation (7), the matrix A^T is the transpose matrix of the matrix A . Through a series of derivations, it is known that the matrix equation (7) has 3 equations whose numbers is equal to numbers of unknown quantities. Therefore, we can get a unique solution about X_1 , X_2 and X_3 .

Based on the matrix equation (7), we wrote the programs of the software for cement-meal batching computation with Visual Basic computer language [4]. Thus, we can make the software respond to errors of the three modules KH, SM and IM, automatically. When using the software, the users can easily and conveniently select or adjust errors among the three modules above mentioned through inputting different values of weights R_1 , R_2 and R_3 .

2. HOW TO PREVENT INPUT MISTAKES TO THE MOST EXTENT

As we described above, cement-meal are formed of different raw materials, as well as ash of fuel coal burned. Actually, every raw material has its relative stable compositions. Through our long-term study and experience, we can give every input parameter a reasonable range. When the software runs, it will show a suggestion on the monitor screen to remind the users to correct input error or possible input mistake or possible sampling faults, if any input parameter exceeds its given range. By the way, if the input parameter exceeding given range is also reasonable or it doesn't affect output results much, you can also ignore it. Thus, this software can prevent or avoid any possible input error or any possible input mistake to the most extent. This is particular important when users carry out a technical design in cement industry.

3. HOW TO GIVE THE EXPERTISE SUGGESTIONS WHEN REQUIRING

Actually, raw materials, which can be used for making cement, are diverse, especially, in modern society, cement industry becomes a kind industry that can use or process some industrial wastes. Whereas, not all raw materials can form the cement-meal that can be fired into the cement product you require. Through studying, we know that when the circumstance, at which raw materials cannot form the cement-meal you need, occurs, there will be a negative number or some negative numbers among the computation results of X_1 , X_2 , X_3 and the computation values of KH, SM and IM. This studying result gives us an inspiration and an idea. If we take all the circumstance, at which any negative number occur, into account, and put corresponding suggestions in the exact positions of the computer programs, it will make the software for cement-meal batching computation give exact suggestions like a cement manufacture expert. These suggestions can help users to make right decision to change some raw materials or to add some corrective raw materials in order to produce the ranked

cement product.

According to our thoughts how to develop the software for cement-meal batching computation as well as some feedback comments from the software's users, we may say this software for cement-meal batching computation, as an important knot in intranet computer controlling system of cement plants and a key in modern technical design processes for cement plants, is a kind of smart software with certain expertise effects, which can play an important role in cement manufacture industry as well as in the process of cement technical design.

4. REFERENCES

- [1] Bai Chonggong, "A Computer Computation Method for Cement-Meal Batching Computation ", Cement (China), No.12, 1985, pp.20 – 23 (in Chinese)
- [2] Li Jianli, Cement Technology. Wuhan: Press of Wuhan University of Technology (China), 1999 (in Chinese)
- [3] Liu Duxin, The Modules Formulae Methods for Cement-Meal Batching Computation, Beijing: Publishing House of China Building Materials, 1992 (in Chinese)
- [4] Liang Puxuan, New Course of Visual Basic Program Designing, Beijing: Publishing House of Electronics (China), 2003 (in Chinese)



Jiang Hongzhou, an associate professor in the Materials School of WUT (Wuhan University of Technology) Wuhan, China and a permanent council member of Thermophysics Society of Hubei province, China. He graduated from WUT in 1985, Master degree, specialty of thermal process in materials production. He was trained in a project of JICA programs in Japan, in 1995. He has published nearly 30 papers and

attended two international conferences, whose papers were compiled in respective proceedings. Some papers were tipped in EI, ISTP and SCI. His research focus is on computer computation and digital simulation in material production processes.

A New Simulation Method Using Multithreading for Modeling Parallel Operated Systems

Wing-Cheong Kwong

Department of Multimedia and Internet Technology,
HK Institute of Vocational Education (Tsing Yi)
Hong Kong Special Administration Region, PRC
Email: wckwong@vtc.edu.hk Tel.: 852-24368704

ABSTRACT

This paper summaries a new method using multithreading and timed Petri Nets (TPN) in modeling and controlling a Parallel Operated System (POS) for Internet users. In addition, the standard clock method is used to trigger the event in the TPN model. The POS in the Laboratory Centre consists of a central controller, an automatic storage and retrieval system with input and output buffers (ASRS), an automatic guided vehicle (AGV) and two Computer Numerical Controlled (CNC) machines with input and output buffers which are represented in respective objects. In Java programs, class members are built with attributes and methods to represent the individual components of the POS. Each of these objects is implemented by making an instance of a respective class member and running it as one of the multiple threads. Coordinating actions between different objects is essential and it is often the case that these activities are time-related. Simulations of actions between these objects benefit by threading because they often model autonomous and interacting entities. The advantages of the proposed method of modeling the POS are that the simulation models can simultaneously provide control capabilities as well as implement the scheduling strategy. The users can use the simulation models through the Internet because the modeling method is relatively simple and the simulation is comparatively fast in Internet applications.

Keywords: Multithreading, Parallel Operated Systems, Timed Petri Nets, and Object-Oriented Programming.

1. INTRODUCTION

A new modeling method using TPN and multithreading for simulating as well as controlling a POS for Internet users is proposed. The modeling method should provide control capabilities as well as implementing the scheduling strategy; the simulation models should also be used through the Internet. Simulation has been used for real-time scheduling of POSs by many researchers. The common framework of a simulation based real-time POS scheduling model is accomplished by linking a simulation model physically with a real POS, with the simulation model working as a monitor [1]. The states of the various elements of the POS will continuously refresh the respective states of the simulation model. However, these simulation-based frameworks for real-time POS scheduling are far from practical [2]. First, a typical POS simulation task requires the evaluation of multiple scheduling policies. It is often found that the simulation is excessively slow when the complexity of the POS increases and thus the degree to which truly real-time decision making is possible is questionable and makes the simulation for Internet use impossible. Secondly, the research work to date has always used the discrete event method for the simulation of the POSs, which cannot

simultaneously provide control capabilities. Chong Peng and F. Frank Chen successfully explored a framework for POS real-time control and scheduling and implemented it by using the Colored Petri Net (CPN) models [2]. The framework uses the strength of the standard clock (SC) technique and ordinal optimization. However the real-time scheduler activation and the simulation monitoring have not yet been tackled and the CPN syntheses method is still not yet fully developed. In addition, the framework for POS real-time control and scheduling is not ready for implementation in an industrial setting. This paper proposes a new modeling method using TPN and multithreading in simulating as well as controlling a POS with the standard clock approach. The users can use the simulation models through the Internet because the modeling method is relatively simple and the simulation is comparatively fast.

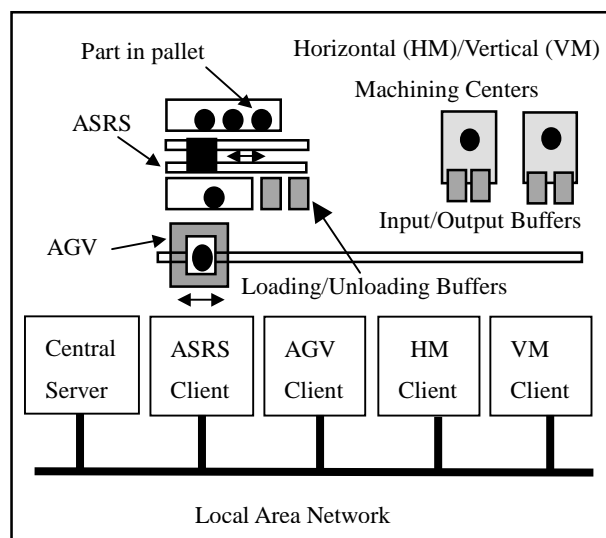


Figure 1.1 The layout of the POS

The layout of a real example of the POS is shown in Figure 1.1. It includes a Central Server, the ASRS, the AGV, and two CNC machines with input and output buffers. Each element of the POS is controlled by a client computer and is connected to the Central Server by a local area network using the Ethernet. Kurapati Venkatesh and Meng Chu Zhou developed the POS control software by object modeling technique diagrams and Petri Nets [3]. In this way it is therefore possible to view all elements of the POS in Figure 1.1 as objects and its control computers also as objects. Traditionally the function of POS control software is to co-ordinate and control different elements in a manufacturing system. Recently, efforts were made to integrate the control software and simulation software in order to expedite system development. Each object such as the Central Server, the AGV Client Computer, the AGV, the ASRS Input and Output Buffers, the CNC Machine and its

associated objects interact with each other to complete a set of production tasks. By using the object oriented design concept, the properties and behavior of the objects of the POS are modeled by the data or attributes and methods or operations of the corresponding software object. For example in Figure 1.1, the MH/VM Client Computer object has properties such as speed of transmission, packet size, and behavior such as commanding the CNC Machine and its input and output buffers and communicating with the Central Server. For modeling the POS, the classes for the real objects of its elements have first to be constructed to incorporate the appropriate properties and behavior. Each of these objects is implemented by making an instance of a respective class member and running it as one of the multiple threads.

For modeling the POS shown in Figure 1.1, the timed Petri Net Model is used to resolve the co-ordination, conflicts, synchronization as well as controlling the elements. There are two common ways to represent the actions of its simulated objects with respect to real or simulated time. In one approach, the clock runs in its usual manner and the standard clock (SC) method is used to evaluate the performance of a set of scheduling rules for the POS. Alternatively, the clock can be moved forward according to the time at which the next action will take place. Kuo, Huang and Yeh [4] used this discrete event driven method to model the dynamic activities in a POS. In this paper, the standard clock method is used to simulate the co-ordination of different elements of the POS in producing different parts. A multi-threading method, which is a powerful capability in Java, employing the standard clock technique has been used to simulate the synchronous activities of the elements in the POS. Multi-threading is the ability of a single process to spawn multiple, simultaneous execution paths. The execution contexts of each thread share the same memory and thus make sharing data between threads simpler than sharing data between processes. Each object running in a thread is created to simulate the operation of each element in the POS and they are run simultaneously to simulate the operation of the complete POS. Each thread can be run in the client computer for each element of the POS respectively. The objects for the elements of the POS can also be run as an applet so that the users in remote locations can run the applets in each client computer through the Internet with Internet Explorer or Netscape Navigator. As a result, the users can use the Internet to simulate the operation of the POS in one place and give orders to the Central Server of the POS in different locations if required.

In section 2 the concepts of the standard clock method and multithreading are discussed. The POS are described in section 3 and the proposed. The timed Petri Nets model of simulation method for controlling and scheduling is elaborated in section 4. The conclusion is given in section 5.

2. STANDARD CLOCK SIMULATION AND MULTITHREADING

Standard Clock Simulation

The SC approach can be illustrated by using an M/M/1 queue simulation example [5] with arrival rate λ and service rate μ . In terms of a stochastic timed automation model, both lifetimes of the event's arrival and departure are exponentially distributed; the parameter for an arrival event "a" is λ , and the parameter for a departure event "d" is μ . In addition to a

nominal (λ, μ) sample path, two or more sample paths can be constructed with parameters $(\lambda + \Delta_1, \mu)$ and $(\lambda + \Delta_2, \mu)$ with $\Delta_2 > \Delta_1$. Suppose all three paths start out at state 0, the triggering events over three sample paths in parallel can be determined by checking

$$\begin{aligned} U &> \lambda/(\lambda + \mu) \quad \text{for path 0,} \\ U &> (\lambda + \Delta_1)/(\lambda + \Delta_1 + \mu) \quad \text{for path 1,} \\ \text{and } U &> (\lambda + \Delta_2)/(\lambda + \Delta_2 + \mu) \quad \text{for path 2} \end{aligned}$$

where U is a randomly generated number. The states of different paths are then updated. In the SC method, the next-event time and type generated by the clock mechanism can be broadcast to multiple state update mechanisms (paths 0,1,2 and so on) which represent the discrete event systems. If the next event received by a special state update mechanism is considered impossible for the DES represented by this state update mechanism, this event will be ignored by the state update mechanism. This triggering event method is used in the simulation of the proposed modeling of the POS.

Multithreading

The object modeling technique diagrams and Petri Nets can be used to model the POS. The elements of the POS are represented by objects which are instantiating the relevant classes that are created in Java programs. Java is unique among popular general-purpose programming in that it makes concurrency primitives available to the applications programmer [6]. The programmer specifies that applications contain threads of execution, each thread designating a portion of a program that may execute concurrently with other threads. This capability, called multithreading, gives the Java programmer powerful capabilities. Every Java applet or application is multithreaded. Every Java thread has a priority in the range of 1 to 10. Some Java platforms support a concept called time slicing and some do not. Without time slicing, each thread in a set of equal-priority threads runs to completion. With time slicing, each thread receives a brief burst of processor time called a quantum during which that thread can execute. There is a Java scheduler and its job is to keep the highest-priority thread running at all times and if time slicing is available, to ensure that several threads with equally high-priority each execute for a quantum in a round-robin fashion. Java uses monitors to perform synchronization. Every object with synchronized methods has a monitor which allows one thread at a time to execute a synchronized method on the object. Other objects in the form of a thread which access the synchronized method without the monitor must wait until the object running with the monitor or which is obtaining the lock finishes executing and releases the monitor or lock. The other way to synchronize the operations of different objects is to use the same stack object. The built-in synchronized methods in the stack object ensure that only one thread accesses the stack object at a time. The proposed modeling method for the POS uses the stack object for synchronization of the execution of the objects which represent the elements of the POS.

3. TIMED PETRI NETS FOR MODELLING OF THE POS

CNC Machine and AGV Classes

Peng and Chen [2] have used colored Petri Nets to develop the object for a typical CNC machine. Using a similar approach, the timed Petri Nets model for one of the CNC machines in

the POS is shown in Figure 3.1. Associated with the CNC Machine (HM) Class, there are other classes such as the CNC Machine Computer, Input Buffer and Output Buffer. The objects of the CNC Machine Computer, Input Buffer and Output Buffer are instantiated from its classes and run as one of the multithreads. The TPN for the CNC Machine consists of ordinary places as in Table 3.1, indicating the status or marking of the machine and its associated elements such as the Machine Computer, the Input Buffer and the Output Buffer.

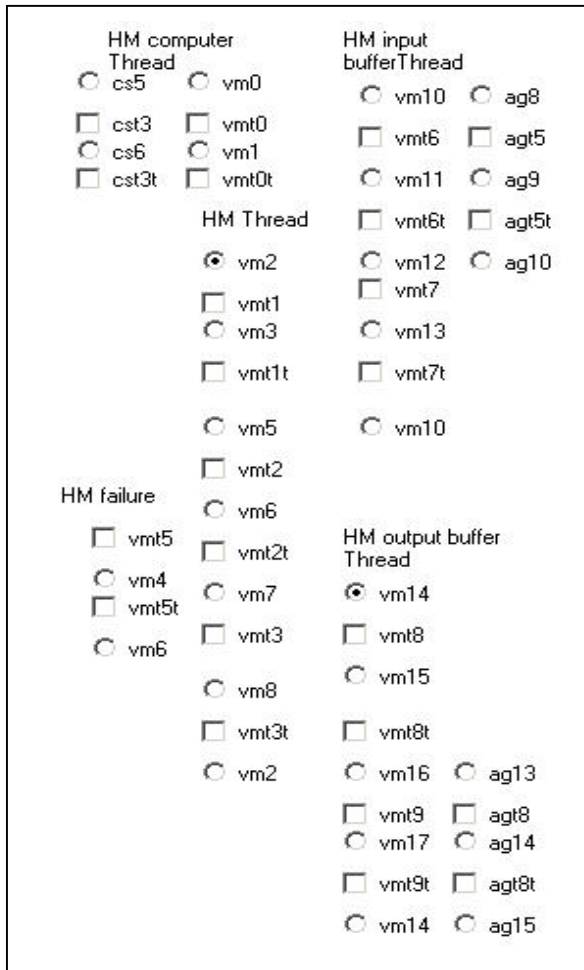


Figure 3.1 Petri Nets Model of the CNC Machine and associated Objects

The CNC Machine Input and Output Buffer Objects have to communicate with the AGV Object which is created in a similar manner. The AGV can either move to ASRS to remove a new part and then move to the machine, or move to the machine to remove the finished part and then move to the ASRS for storage. Some relevant places (such as place ag8 at which AGV with a new part is at the machine input buffer) and transitions (such as transition agt5 at which AGV begins loading the machine input buffer) of the AGV Object are duplicated at the appropriate places (such as place vm10 at which the machine input buffer is available) as well as the transitions (such as transition vmt6 at which the input buffer begins loading from AGV) of the Input and Output Buffers Objects as in Figure. 3.1. The Machine Computer Object is also created and its tasks are to communicate with the Central Server Object for updating the states of the CNC Machine

Object and its associated objects to the Central Server. At the same time, the Machine Computer Object will receive the commands and other necessary information from the Central Server and give instructions to the CNC Machine, Input and Output Buffer Objects. The information is stored in a common Stack Object for various Objects and this insures that only one Object running as a thread accesses the Stack Object at a time and so that the synchronization of the operations of the Objects is achieved.

Table 3.2 Entities for the timed Petri Nets of the CNC Machine Objects

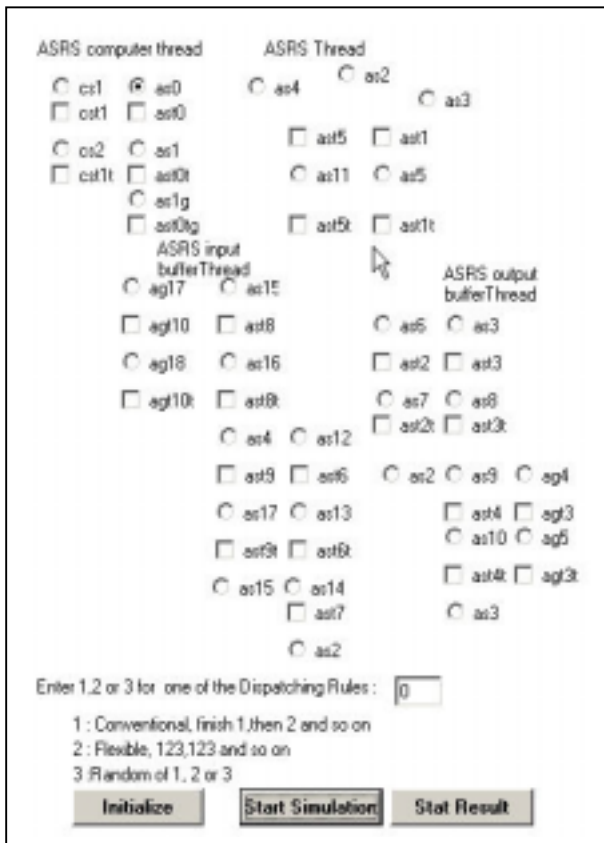
Places	Remark	Transitions	Remark
TPN for CNC Machine (HM) Object			
vm2	Machine Available	vmt1	Begin to load from the input buffer
vm3	Part loading into machine	vmt1t	Finish loading from the input buffer
vm4	Machine failure	vmt2	Begin machining the part
vm5	Part in machine	vmt2t	Finish machining
vm6	Processing part by machine	vmt3	Begin loading into the output buffer
vm7	Finished part in machine	vmt3t	Finish loading into the output buffer
vm8	Loading of part into output buffer	vmt5	Machine begins to fail.
		vm5t	Machine has been repaired
TPN for HM Input Buffer Object			
vm10	Input Buffer Available	vmt6	Begin loading into the input buffer
vm11	Part loading into the input buffer	vmt6t	Finish loading into the input buffer from AGV
vm12	Part at the input buffer	vmt7	Begin loading into the machine
vm13	Part loading into the machine	vmt7t	Finish loading into the machine from the input buffer
TPN for HM Output Buffer Object			
vm14	Output buffer available	vmt8	Begin loading into the output buffer
vm15	Part loading into output buffer from machine	vmt8t	Finish loading into output buffer from machine
vm16	Finished part at the output buffer	vmt9	Begin loading into the AGV
vm17	Part loading into the AGV	vmt9t	Finish loading into the AGV from the output buffer
TPN for HM Machine Computer Object			
vm0	Machine Computer ready to communicate	vmt0	Begin communicating with Central Server
vm1	Machine Comp. communicates with Central Server	vmt0t	Finish communicating with Central Server

The Input Buffer, Output Buffer and AGV Objects are running as threads and synchronization between these objects is required to simulate the operations of the POS. The transitions in the objects represent the start or the termination of an event

in the TPN for that object. When a transition is enabled, it is fired immediately, no time delay is associated with the transition. The firing of a transition means that the tokens with the highest priority in the transition list according to the standard clock method are consumed and new tokens are generated by the transitions.

ASRS and Central Server Classes

The class for the TPN model shown in Figure 3.2 of the ASRS Object in the POS is constructed in a similar manner to that of the CNC Machine class. It also has Input and Output Buffers Object that are instantiated from its classes. The ASRS Computer Object is also created and its tasks are updating the states of the ASRS Object and its associated objects to the Central Server. At the same time, the ASRS Computer Object will give instructions to ASRS, Input and Output Buffer Objects. The operations of the ASRS Object also have to synchronize with that of the AGV Object. For example, the ASRS Input Buffer Object is empty (place as4) and the AGV Object has a part and is at the ASRS input buffer. The execution of these two Objects must trigger the transitions at “ast7” (begin to load a part) and at “agt10” (begin to unload from AGV) concurrently. The synchronization of the operation of different objects is achieved by accessing the same Stack Object.



initialisation of the Central Server Object,
CentralServer(Form1 bform, Stack aStack, Stack bStack, Stack cStack, Stack dStack, Stack eStack, Stack fStack, Stack gStack, Stack hStack)
{CentralServerForm = bform;
The_ASCMIB_Stack = aStack;
The_ASCMOB_Stack = bStack;
The_AGVCM_Stack = cStack;
The_VMCMIB_Stack = dStack;
The_VMCMOB_Stack = eStack;
The_AS_ASRS_Stack = fStack;
The_AS_AGV_Stack = gStack;
*The_AS_VM_Stack = hStack; } / * Stacks for storing status*
information of and commands for the POS/*

Construct the method or functions for the operations of the Central Server Object :

```
public void run() { /* start running of the Object */
// Check initial conditions of the POS
while (!CentralServerForm.quittingTime) {
/* check whether quittingTime is true */
/* Start communication with the threads of computers of ASRS,
AGV and VM */
/* Start communicating with ASRS computer */
if (CentralServerForm.cs1.getChecked() &&
CentralServerForm.cs1.getChecked() &&
CentralServerForm.as0.getChecked() &&
CentralServerForm.CStick )
{ /* place cs1 to cs2 at which the Central Server computer is
communicating with ASRS computer */
CentralServerForm.CStick = false;
/* Wait for next CLKtick derived from next possible event
from the standard clock method */
/* Synchronizing with ASRS computer */
The_AS_ASRS_Stack.push(new Integer(PartNo)); /* load
information into the stack for ASRS computer */
while (The_AS_ASRS_Stack.size() == 0) /* Central Server
Object is waiting for ASRS Computer Object */
CentralServerForm.cs1.setChecked(false);
/* leave the current place */
CentralServerForm.cs1.setChecked(false);
/* switch off the current transition */
CentralServerForm.cs2.setChecked(true);
/* Enter a new place */
System.out.println("ASRS Computer is responding and
updating its tokens");
/* Take about 0.1 seconds for transmitting data */
try { Thread.sleep(SetTime/CentralServerForm.factor); }
catch (Exception f) { }
/* Update the states of the ASRS Object and the places of
AGV, VM remain as before updated */
CentralServerForm.as2UD =
CentralServerForm.as2.getChecked(); /* ASRS crane ready */
CentralServerForm.as3UD =
CentralServerForm.as3.getChecked();
/* ASRS output buffer empty */
CentralServerForm.as4UD =
CentralServerForm.as4.getChecked();
/* ASRS input buffer loaded */ /* Continue communication for
the AGV and Machine Computer Objects */
```

Implementing the scheduling rule :

```
switch (CentralServerForm.dispatchRule) { /* select the
dispatching rule */
case 1 : /* dispatchRule */
/* Conventional: finish part1, then part2, and then part3 */
```

```
if (i1 > 0) PartNo = 1; else if (j1 > 0) PartNo = 2;
else if (k1 > 0) PartNo = 3; else PartNo = 0;
switch (PartNo) { /* Conventional */
case 0: System.out.println("No more new part to be
produced."); break;
case 1: /* Update information in case 1 into the Stack */
Integer anInt1 = new Integer(PartNo);
The_ASCMOB_Stack.push(anInt1);
anInt1 = new
Integer(CentralServerForm.part1TimeM1);
The_ASCMOB_Stack.push(anInt1); i1--; break;
case 2: /* Update information in case 2 into the Stack */
Integer anInt2 = new Integer(PartNo);
The_ASCMOB_Stack.push(anInt2);
anInt2 = new
Integer(CentralServerForm.part2TimeM1);
The_ASCMOB_Stack.push(anInt2); j2--; break;
case 3: /* Update information in case 3 into the Stack */
Integer anInt3 = new Integer(PartNo);
The_ASCMOB_Stack.push(anInt3);
anInt3 = new
Integer(CentralServerForm.part3TimeM1);
The_ASCMOB_Stack.push(anInt3); k3--; } /* switch
(PartNo) for conventional dispatching rule */
/* Continue for deciding other dispatching rule */
```

Implementation of the Method

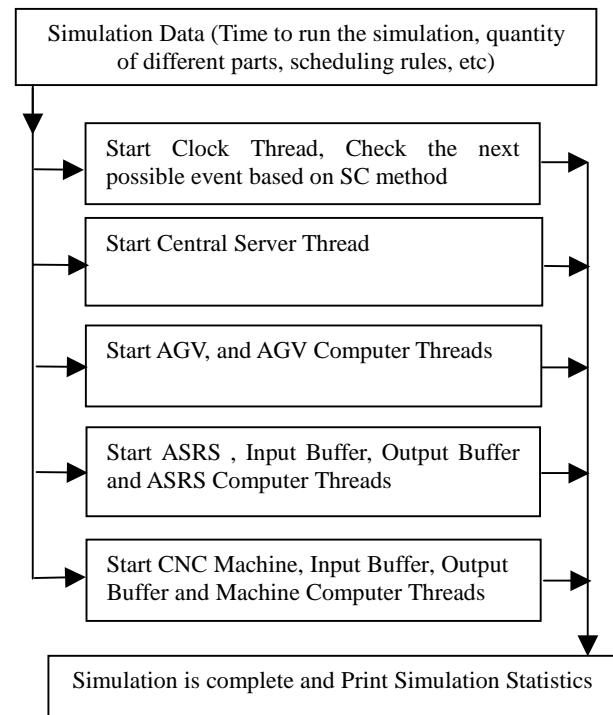


Figure 4.1 Flow Diagram of the Simulation program for the POS

The Clock Object was instantiated from the Clock class and was implemented as the runnable thread is created. The Clock thread is an endless loop that updates the simulation time every second. It functions as the standard clock and no state information of the elements of the POS is necessary for updating the clock mechanism since the state update and the clock mechanisms are completely decoupled. As a result of this decoupling, the next event time and the type of the places generated by the clock mechanism can be broadcast to

multiple state update mechanisms, which represent the Petri Nets model of the POS. The flow diagram of the simulation program is shown in Figure 4.1. The program is written in Java using the Visual J++ 6.0 Development Tool. The Classes for Clock, AGV and AGV Computer, ASRS, ASRS Computer, ASRS Input Buffer and Output Buffers, CNC, CNC Machine Computer, CNC Input and Output Buffers, and Central Server are created and are either implemented as the runnable threads or extended as threads. The scheduling rules are entered into the Central Server Object and the simulation process begins. The Clock Object thread starts and the rest of the simulation revolves around time-stepping through the total simulated time. The triggering of the events in various objects of the POS is determined by the standard clock method where the probability of firing the transition is larger than the random generated number "U" and the clock moves forward in the usual manner until the simulation is complete due to either the end of the time for simulation or the finish of the parts to be produced.

All the classes for the elements of the POS are created according to the methods proposed in session 3 and they are either extended as threads or implemented as runnable threads. The multi-threaded objects that are instantiated from the classes are running asynchronously and autonomously and are changing its states in the places of these classes. In this approach, the objects are sharing data and co-ordination as well as synchronization between the objects of the POS can be achieved. They can also be run as applets and derived for the control software for the elements of the POS. The multithreaded objects of the POS are tested and the simulation result for producing two different parts in one CNC machine is shown in Table 4.1. It can be seen from the simulation result that the utilization of ASRS, AGV and CNC Machine are 34%, 29% and 69% respectively.

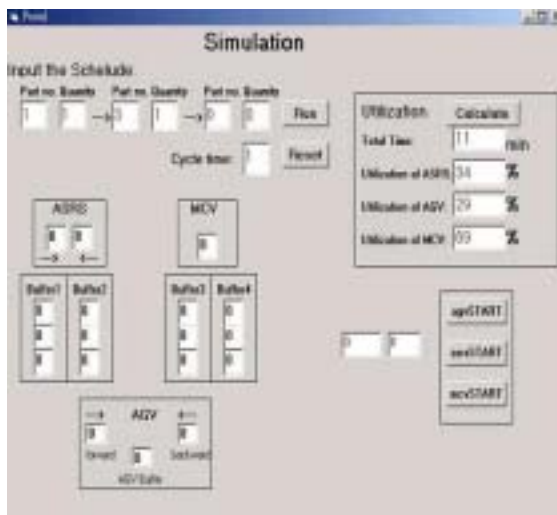


Table 4.2 Result of producing two different parts by the proposed simulation model

5. CONCLUSIONS

Normally the Timed Petri Nets (TPN) models are useful in investigating qualitative or logical properties of concurrent systems and are ideal for discrete event systems but are not possible for the real-time control of the POS. The TPNs become very complicated to implement when the complexity

of the POS increases. Hence this approach is not applicable when the simulation is put up on the server at the web site and the access to the simulation is through the Internet. Instead of using this conventional discrete-event scheduling method, the SC method is employed and the elements of the POS are represented as objects that are instantiated from the respective classes. In the SC method, no state information is necessary for the update mechanism since the clock mechanism and the state updating mechanism are completely decoupled. The complexity of the model of the POS using multithreads will thus decrease and the simulation model becomes simple and easy to understand. The simulation model using threads and applets has been used to simulate the production of the POS for a small batch size of parts and it is possible for the users to find the utilization of various elements of the POS through the Internet. It has been proved to be possible that the proposed simulation model using TPN models and multithreaded objects with the standard clock approach can describe the performance of the POS with respect to throughput, buffer stations, deadlocks and ease of operation under different scheduling rules for the different parts.

6. REFERENCES

- [1] G.M. Harmonosky, "Implementation issues using simulation for real-time scheduling, control, and monitoring", Proceedings of 1990 Winter Simulation Conference, pp. 595-598, 1990.
- [2] Chong Peng and F. Frank Chen., "Real-Time Control and Scheduling of Flexible Manufacturing Systems: An Ordinal Optimisation Based Approach", The Int. Journal of Advanced Manufacturing Technology, 14: 775-786, 1998.
- [3] Kurapati Venkatesh, Meng Chu Zhou, "Object Oriented Design of FMS Control Software Based on Object Modeling Technique Diagrams and Petri Nets" Journal of Manufacturing Systems, Vol. 17/No. 2, 1998.
- [4] Chung-Hsien Kuo, Han-Pang Huang and Min-Chin Yeh, "Object Oriented Approach of MCTPN for Modelling Flexible Manufacturing Systems.", The Int. Journal of Advanced Manufacturing Technology, 14: pp. 737-749, 1998.
- [5] Christos G. Cassandras, "The Standard Clock Approach", Discrete Event Systems: Modelling and Performance Analysis, Aksen Associates Incorporated Publishers, pp692-699, 1993.
- [6] Deitel and Deitel, Java : How to Program, Prentice Hall, pp734-773, 1999.



Kwong Wing-cheong is a Senior Lecturer in the Department of Multimedia and Internet Technology of the Hong Kong Institute of Vocational Education (Tsing Yi) of Vocational Training Council. He graduated from University of Westminster, UK in 1982 with B.Sc(Eng.) in Instrumentation and Control Engineering; from The Hong Kong Polytechnic University, HK in 1995 with M.Sc. in Industrial Automation. His research

interests are in distributed parallel processing, parallel operated systems and multi-players networked games.

The Design and Implementation of a Multi-Auctioneer Prototype System for Grid Resource Management

Xiu-chuan Wu^{1, 2}, Hao Li³ and Jiu-bin Ju¹

¹School of Computer Science & Technology, Jilin University, Changchun, Jilin Province 130012, China

²School of Computer Science & Technology, Yantai University, Yantai, Shandong Province 264005, China

³School of Computer Science & Technology, Jilin Normal University, Siping, Jilin Province 136000, China

Email: wxc225@163.com Tel.: 13604309089

ABSTRACT

From the viewpoint of resource management for computational grid, computing economy method is very appropriate and effective. A multi-auctioneer prototype system based upon computing economy method for grid resource management is designed and implemented in this paper. The prototype system has following advantages: Firstly, it eliminates the bottleneck of job scheduling when there are abundant jobs that are needed to submit to the grid system; secondly it prevents from the dishonest auctioneer's fraud action. The system provides mechanisms for optimizing resources provider and consumer objective functions.

Keywords: Computational Grid, Resource Management, Computing Economy, Auction, Auctioneer

1. INTRODUCTION

The accelerated development in grid computing and Peer-to-Peer (P2P) are emerging as a new paradigm for solving large-scale problems in science, engineering and commerce^[1]. Resource management is one of the most important contents for computational grid. Because the dynamic performance behavior as well as heterogeneity of resource of the computational grid, system have no integrity information about resource, resource management is a well-known difficult problem. There are many kinds of management methods and these methods also refer to resource reservation and rescheduling and QoS and so on. Besides that network resources are included into computational grid, so systems boundary problem must be handled for resource scheduling system and also a need for a distinctive security and fault-tolerance mechanism.

Generally speaking the resource management system of the computational grid decides the validity and acceptability in the large degree. Resource management is composed of resource dissemination and resource discovery and task scheduling as well as QoS and so on. The methods of the task scheduling also affect the structural model of the resources management system. The structure of the scheduling program relies on the resources number that can be obtained by it and the domain that the resources are in. The resources management system also relate to the organizing framework of the machines. Besides that the resources management system bear on the resource model for it decides the application and the resources scheduling system how to describe and manage the resources of the computational grid.

There are four methods for resource management for

computational grid^[1]. They are respectively hierarchical management model, computing economy management model, and abstract owner management model and blend management model and so on.

In the centralized environment of the hierarchical management model method all the resources of the computational grid are scheduled by a central instance. It is adapt to the management of cluster management system and batch job queue system such as in the calculating center for the reason that all the resources can be used in the same objective. Another advantage is it can be neglected even if in the case of lacking network bandwidth. Obviously it is the bottleneck of the scheduling system. Invalidation of it will result in the collapse of the whole system. In the distributed model each scheduling program interacts and submits jobs to the remote system. The allocation and scheduling of the resources were decided by both of the resources requester and the resources provider. The method overcomes the disadvantages of the central model but it is very difficult for supporting for the applications of the multi-site and it is also very difficult for the synchronization of the jobs and assuring the execution in the same time. Even though using pull or push operation cannot improve the performance. Hierarchical scheduling model is a central scheduling model in nature because there is an independent scheduling program that is used for scheduling native resources. Some systems such as Globus use this kind of scheduling model. In fact this system is only an interface to the native resources. Just like in a real world market, there exist various economic models for setting the price of services based on supply-and-demand and their value to the user. In the computational grid this kind of model is called computing economic model and it is most suits for a computational grid system. Abstract owner model and blend model are so complicated that they are only theoretical model and they are not used in practice.

This paper analyzes and researches deeply the economic models for resource management and task scheduling in grid computing, and then designed and implemented the resources management system of the multi-auctioneer for computational grid. Simulating experimental results shows the resources scheduling system eliminates the bottleneck of job scheduling when there are abundant jobs that are needed to submit to the grid system and as a result improves the performance of grid system. Furthermore users can select an auctioneer from multiple-auctioneers so prevents from the dishonest auctioneer's fraud action. The system provides mechanisms for optimizing resources provider and consumer objective functions.

2. COMPUTING ECONOMIC MODEL FOR RESOURCE SCHEDULING

Literature [2][3][4][5][6][7] and so on bring forward the concept of the computing commerce i.e. G-commerce or computing economy. It uses economy principle that comes from the human being market to fulfill the management and scheduling of resources in the processes of computing. It is called computing economy. The computational grid is as the product-oriented (commerce) computing environment. In this environment the resources owner is called producer (Vendor) and the user is called consumer (buyers). Computational grid environment takes different resources as interchangeable commodities. The result is both of providers and requesters of the resources drive the price of the resources. The both of aspects want to maximize their proceeds. The model of the resources management and task scheduling system of the computational grid based on the economy has distinct advantages. The main characteristic is adjusting the contradiction of the supply and demand. Because the processes of deciding the stratagem of resources scheduling is distributed to the users and the resources owners so the results are change system as center to users and the users can decide by themselves to obtain the best capability only pay out the least cost. Furthermore it can also help the developer to develop scheduling stratagem and to come into begin the high extensible system.

In the computing economic models the owner of the resources can build the service price according to the demands of the consumers by resources brokers of the computational grid and it can also to issue the services that can be provided by it to the users out of the world by market catalog of the computational grid. It also can issues other information to users such as accessing price at the different period of time and some kind of discount at the non-peak value and so on.

Generally speaking the standards of the evaluating computing economic model are prices stability, market equilibrium, application and resources validity and so in the scope of computational grid. For the validity of insuring scheduling prices stability is the key factor. Obviously, the application and scheduling based on the price fluctuate as price fluctuate very much. Furthermore bring on the result of performance to descend. In addition equilibrium is a degree to measure if the price equitable. If the totally market cannot import the equilibrium the corresponding cost and the treated value cannot be trusted so the system is abortive.

At present there are many of economic models are used for resources scheduling in the computational grid. These models include Commodity Market Model, Posted Price Model, Bargaining Model, Tendering/Contract - Net Model, Auction Model, Bid - based Proportional Resource Sharing Model, Community/Coalition/Bartering Model and Monopoly and oligopoly Model and so forth.

Most common used economic model is auction model. For in auction model comparatively small price information is needed and on the other hand it is also implemented easily relatively. An auction is a market institution with an explicit set of rules determining resources allocation and prices on the basis of bids from the market participants.

The types of auctions to be used are English Auction (first-price open cry), First-price sealed-bid, Second-price sealed-bid (Vickrey), Dutch (descending) and so on.

Auctions are stylized markets with well-defined rules, modeling them is very appropriate. Moreover, several of the motivations behind auctions are similar to the motivations behind the asymmetric information contracts. Beside the mundane reasons such as speed of sale that make auctions important, auctions are useful for a variety of informational purpose. Often the buyers know more than the seller about the value of what is being sold, and the seller, not wanting to suggest a price first uses an auction as a way to extract information. Art auctions are a good example, because the value of a painting depends on the buyer's tastes, which are known only to him. Auctions are also useful for agency reasons. Because they hinder dishonest dealing between the sellers's agent and the buyer.

Auctions can be differentiated across many parameters including, but not limited to, those concerning: matching algorithm, price determination algorithm, event timing, bid restrictions, and intermediate price revelation and so on^[8,9]. Generally speaking these models allow relativity of values between the multi-bidders and also include private values model and public value models^[8]. Assume there are n bidders. X_i denotes private signal of being observed by bidder and suppose $X=(x_1, x_2, \dots, x_n)$, $S=(s_1, s_2, \dots, s_m)$. S is the fact of quality for measuring such item. For S are not seen by any all other bidders. Now we suppose the $v_i(s, x)$ is the value of the bidder i . Thus values of any bidders are not relying on the private signals but also other signals, which can not seen by itself i.e. private values of any other bidders and public value^[9]. When $m=0$ at the same time $v_i=x_i$ for all i this model is changed into the independent private value model. When $m=1$ and $v_i=s_i$, the model is changed into public value model. The two factors for random variable $v_i(s, x)$ are affiliated. If one variable is increased the other variable is also increased i.e. they have the positive relativity.

The analysis of the entire standard model is based-upon the following four supposes^[3,8]:

- 1) all the bidders are risk-litmusless;
- 2) possess the independent private value;
- 3) among all the bidders are symmetrical;
- 4) payment is the function of bid and no cahoots.

3. THE DESIGN AND IMPLEMENTATION OF THE MULTI-AUCTIONEER SYSTEM MODEL

At present there are a lot of resources management systems that based upon auctions model. But all the systems use one auctioneer. This kind of systems exist two distinct faults. Firstly it may be a bottleneck when lots of jobs are needed to schedule. Secondly auctioneer may cahoots with some one bidder to fraud other bidders. In order to overcome these defaults we design and implement the multi-auctioneer system for grid system.

The architecture of the system is as illustrated in Figure 1. In the figure GTS is a Grid Trade Service^[3,4]. Its main function

is to enable resource trading and execution of consumer requests. In this paper GridSim is used for simulating the algorithm. The data structures which are used in the system are described as in the following^[10, 11].

/* the description of consumer for resources request */

```
Class Res_Req_Def
{
    private double Deadline; //deadline set by user
    private double Budget; //budget
    private double StartTime; //set by user or current
    time
    private double EndTime; //EndTime <= StartTime
+ deadline
    public Res_Req_Def(double Deadline,double Budget)
    {
        this.Deadline = Deadline;
        this.Budget = Budget;
    }
}
```

/*resources description provided by Grid Service Provider */

```
Class Res_Def
{
    public String arch; //architecture of the server
    public String opsys; //operating system used by
    server
    public int No_of_PEs; // PE(Processing Element) is
    CPU unit
    public PE_SPEC_Rating; //velocity of computing
    public double Cost_Per_Sec; //fee per second public
    Res_Def (String arch, String os, int No_of_PEs, int
    PE_SPEC_Rating, double Cost_Per_Sec)
    {
        this.arch = arch;
        this.opsys = os;
        this.No_of_PEs = No_of_PEs;
        this.PE_SPEC_Rating = PE_SPEC_Rating;
        this.Cost_Per_Sec = Cost_Per_Sec;
    }
}
```

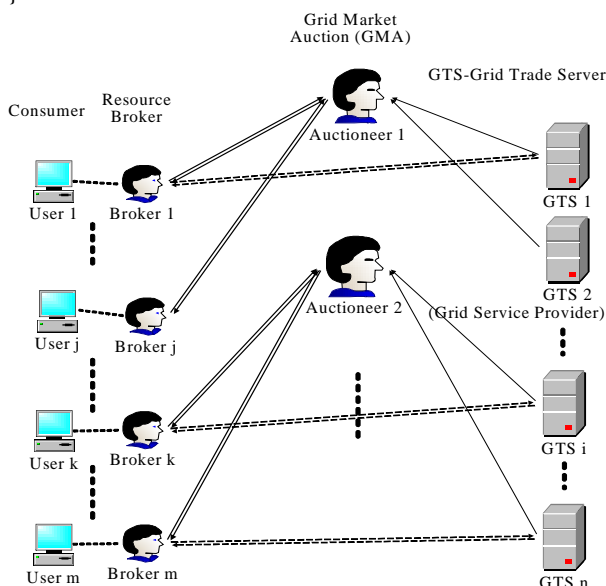


Figure 1 the architecture of multi-auctioneer for resources management

Between the auctioneer and the users is a resource broker. Its

main function is to contact auctioneer for users. We describe algorithm as following:

/* resources broker:

1. receiving the request that comes from the users for resources selection and know the demand of the consumers such as deadline and budget and so forth;
 2. selecting the auctioneer;
 3. bidding to auctioneer for resources after selecting the auctioneer;
 4. if succeed build the affiliation between the consumer and the resource provider;
- */

Based upon literature [10, 11] we write the pseudo code as following:

```
Class Broker
{
    public void SetResReq() //setting user demand;
    {
        Res_Req_Def Req_Set[] = {
            new Res_Req_Def(2*60*60,5000);
        }
    }
    public void GetAuctioneer() //obtain the information
    about auctioneer;
    {
        Auctioneer auctioneers = {
            new Auctioneer(),
        }
    }
}
```

Based on the following standards to select the auctioneer:

1. How many resources and what kinds of resources the auctioneer possess(Res_Available()). If they can meet the demands for the user?
2. Auction Reserve_Price so as to decrease the fee for user as possible as;
3. Based upon degree of trust (for example the amount of strike a bargaining in the past time), so as to select an auctioneer;

Auction strategy, which can be optional such as English auction or Dutch auction. In this paper double auction is used and users are grouped in the way of odd number users for auctioneer1 and even number users for auctioneer2.

For an auctioneer it must provide about resources information provided by some provider. At the same time it also must obtain the demand information provided by bidder and finally to decide the auction strategy. Certainly it must control the process of the auction process and maximize the profit for both provider and consumer.

/*First we describe the data structure of the auctioneer^[10].
*/

```
Class Auctioneer
{
    public static final int Double_AUCTION = 4; // double
    auction model is used in the paper.
    private double P_adjust; //adjusting the price for
    establish reserve price
    public double Reserve_Price; //formula for calculating
    reserve price[11,12];
```

```

/*
Reserve_Price=
  No_of_Res
[  $\sum_{i=1}^{No\_of\_Res} (resource\_set[i].No\_of\_PEs * resource\_set[i].$ 
  PE_SPEC_Rating * resource_set[i].Cost_Per_Sec)] /
  No_of_Res + P_adjust (1)

```

Resource, which is considered in the system, is only CPU. In the formula (1) No_of_PEs represents the all number of cup that is used in the system. PE_SPEC_Rating is computing velocity of processor unit. Cost_Per_Sec means the fee that must be pay by the user for their using to CPU in per second.

```

/*
public int Auction_succ; //degree of trust of
auctioneer
public int No_of_Res; //amount of the valid
resources
private static void GetResDef() //obtain resources
information
{
  Res_Def resource_set[] = {
    New Res_Def("Intel Pentium", "Linux",
2,380, 1),
    ...}
}

```

The realizing process of auction is as following:

```

public int Auction();
public double ResAvailable ();
/* calculating valid resources and according to the resources
amount as well as resources description so as to obtain
calculating formula 2[11]:

```

```

  No_of_Res
Res_Available=[  $\sum_{i=1}^{No\_of\_Res} (resource\_set[i].No\_of\_PEs)$ 
  *resource_set[i].PE_SPEC_Rating] (2)
*/

```

```

public Auctioneer(int No_of_Res, int policy)
{
  this.No_of_Res = No_of_Res;
  this.policy = policy;
}

```

Based on game theory for the double auction model following formula (formula 3) makes resource providers and resource consumers both sides reaching the maximum payoffs. This is a Bayesian Nash equilibrium for this case^[13, 14, 15].

$$p_s(c) = \frac{1}{4} + \frac{2}{3}c \quad (3)$$

$$p_b(v) = \frac{1}{12} + \frac{2}{3}v$$

Thereinto P represents price, c represents cost of resources and v represents the value evaluated for resources by the consumers. S is denotation of seller and b stands for buyer of resources.

When $P_b \geq P_s$, there is a business take places each other and the trading price is $P = (P_b + P_s)/2$ and then this P is as new P. The algorithm of set auctioneer and to bidder is as followings:

a. Set Auctioneer():

```

1. Res_Req = Res_Req_Def.Deadline * Res_Req_Def.Budget;
//resource broker calculate resources needed by user

```

```

2. Auctioneer(i);
3. Res_Available(i) // obtain information about resources;
4. calculating average price  $P_s$  based on formula (3);
5.  $i = i + 1$ ;
6. IF Res_Req > Res_Available(i) THEN goto 2;
7. eval (j) = f(Reserve_Price(i), Auction_succ(i));
//according to Reserve_Price as well as Auction_succ to
//record auctioneer information and evaluate auctioneer
//based on the function f();
8.  $j = j + 1$ ;
9. goto 2;
10. evaluating all the auctioneers and calculating the best
    auctioneer for next user select;
11. goto b. //bid

```

b. Bid():

```

1. policy = Double_AUCTION; // Double Auction is used;
2. provider provide available resources amount
   Res_Available and Reserve_Price;
3. calculating bid value  $P_b$  based on formula (3);
4. IF  $P_b \geq P_s$  THEN
   bargain each other on the price  $P = (P_b + P_s)/2$ 
   and P is as new P for next P.

```

If several bid are same then selecting one user by auctioneer according to the bid time stamp?

4. SIMULATION MODEL AND PERFORMANCE ANALYSIS

In this section we describe briefly the simulation model of our system architecture. As described in section 3, our system can eliminates the bottleneck of job scheduling when there are abundant jobs that are needed to submit to the grid system so that can improve the performance of the scheduling system of computational grid.

We use GridSim^[10] to simulate the resources in the grid environment. Being the focus of this paper on the performance improving of the resource scheduling, we do not simulate different kinds of resources but they are the same machines and the price of these resources all are the same. Moreover we limited the jobs that are submitted to the system are also same. They all do the same work and these works are very simple. They are neither quality-sensitive nor price-sensitive. We also did not consider other factors such as checkpoint setting and reschedule of tasks as well as QoS of grid system and so on.

Experimental results are shown in Figure 2. In figure 2 in single auctioneer system when few jobs (nearly equal the number of resources in the grid environment) are submitted to the system the resources using rate of the resources is higher. Nearly every resource can be used by these jobs. In this case relatively in multi-auctioneer system multi-auctioneer schedules jobs so some time are consumed. When number of jobs increased, because resources are auctioned (scheduled) only by one auctioneer, though there are many idle resources jobs cannot be allocated to those resources. Comparatively multi-auctioneer system can effectively use computing resources in the computational grid. As the result resource using rate are heightened and performance of resource scheduling are improved greatly.

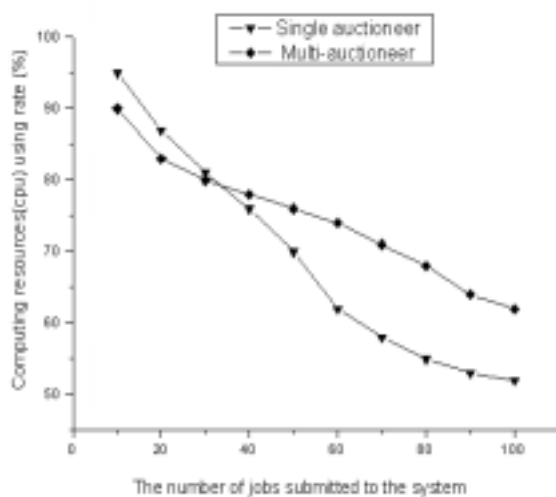


Figure 2 Resource using rate of comparing single auctioneer and multi-auctioneer system

When one of the auctioneers deceives users using high price to purchase resources, users can turn their steps to other auctioneers. Comparing with single auctioneer the mechanism is a bestirring mechanism. In this case of single auctioneer the payoffs both sides are zero. It is shown in Figure 3.

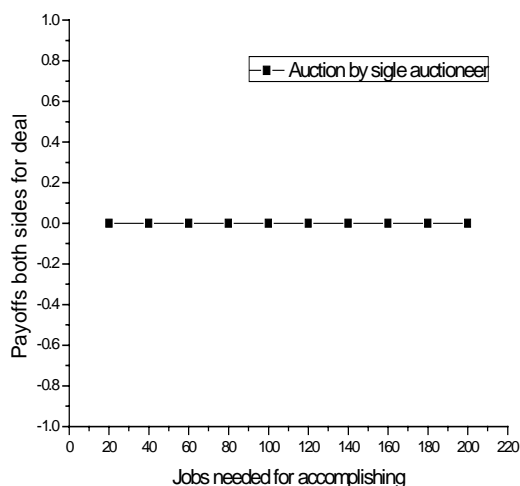


Figure 3 Payoffs both sides when deceiving takes place for single auctioneer

5. CONCLUTIONS AND FUTURE WORK

The multi-auctioneer system for resource management for computational grid system is designed and implemented in this paper. The system has at least two advantages. In the case that there are many jobs are submitted to the grid system it delimits the bottleneck of submitting section. Another advantage is for any user it can select auctioneer among many of them so it can prevent effectively from the auctioneer's fraud action.

At present it is only a simple prototype system. Many factors

do not be considered. These factors include job deadline, cost as well as budget restriction and QoS of network and so on. Next these factors would be considered and the system would become more consummate. On the other hand at present the system can only prevent from the auctioneer's fraud action. Auctioneers cannot find bidder's cahoots behavior. In the future we plan to design the mechanism to solve this problem. We think the factors affecting the cahoots depend on the two aspects, which are rules of auction and characteristic of resources being auctioned. If the bidder cahoots are found then auctioneer can adjust auction mechanism and or increase reserve price or use secluded reserve-price system.

6. REFERENCES

- [1] Klaus Krauter, Rajkumar Buyya, and Muthucumaru Maheswaran. A Taxonomy and Survey of Grid Resource Management Systems. *Software Practice and Experience*, Vol. 32, No. 2, Feb. 2002, pp. 135-164.
- [2] Rajkumar Buyya, Jonathan Giddy, and David Abramson, A Case for Economy Grid Architecture for service - Oriented Grid Computing. 10th IEEE International Heterogeneous Computing Workshop (HCW 2001), In conjunction with IPDPS 2001, San Francisco, California, USA, April 2001.
- [3] Rajkumar Buyya, David Abramson, Jonathan Giddy, and Heinz Stockinger, Economic Models for Resource Management and Scheduling in Grid Computing. Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Wiley Press, USA, May 2002.
- [4] Rajkumar Buyya, Nimrod/G Problem Solving Environment and computational Economics, Grid Computing Environments Community Practice (CP) Document, Global Grid Forum (GGF)/First GGF Workshop, Amsterdam, The Netherlands, March 4-7, 2001.
- [5] R. Wolski, J. Plank, J. Brevik, and T. Bryan. Analyzing Market-based Resource Allocation Strategies for the Computational Grid (UT Tech. Rep. #CS-00-453).
- [6] R. Wolski, J. Plank, J. Brevik, and T. Bryan. G-commerce- market formulations controlling resource allocation on the computational grid. Technical Report UT-CS-00-450, University of Tennessee, October 2000.
- [7] R. Wolski, J. Plank, and J. Brevik. G-Commerce--Building Computational Marketplaces for the Computational Grid (UT Tech. Rep. #CS-00-439).
- [8] Armstrong, Mark. Optimal Multi_Object Auctions. *Review of Economic Studies*. 2000(67),455-481.
- [9] Paul Klemperer. Auction Theory: A Guide to the Literature. *Journal of Economic Surveys*. 1999 Vol.13,no.3.
- [10] Rajkumar Buyya and Manzur Murshed, GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing, The Journal of Concurrency and Computation: Practice and Experience (CCPE), Volume 14, Issue 13-15, Pages: 1175-1220, Wiley Press, USA, November - December 2002

- [11] Rajkumar Buyya. Economic-based Distributed Resource Management and Scheduling for Grid Computing. Ph.D thesis. School of Computer Science and Software Engineering Monash University, Melbourne, Australia. April 2002.
- [12] <http://www.specbench.org/osg/cpu2000/results/cpu2000.html>
- [13] Zhang wei-ying, Game theory and information economics. SHANGHAI RENMIN CHUBANSHE 1996. ISBN 7-208-02432-4, In Chinese.
- [14] Shi xi-quan, Game Theory. SUFEP. 2000. ISBN 7-81049-398-1/F . 334, In Chinese.
- [15] Beirman, H. Scott and Fernandez, Louis. 1998. Game Theory with Economics. Addison-Wesley.



Wu Xiu-chuan is a Ph.D. student of Department of Computer Science at the Jilin University and an association professor of Department of Computer Science at the YanTai University. His current research interest is Distributed Computing System and Grid Computing.



Hao Li is a teacher of Jilin Normal University. He got M.Sc degree from Jilin University. His current research interest is Distributed Computing System and Grid Computing.

Ju Jiu-bin is a professor of School of Computer science & Technology, Jilin University. His current research work includes Distributed System & Network Software.

Research on Resilient Distributed File Systems

Li Zhonghua¹, Li Weihua¹, Zhang Lin²

¹ School of Computer Science, Northwestern Polytechnical University, Xi'an Shanxi 710072, China

² Art Engineering College, Xi'an Engineer Science & technology University, Xi'an Shanxi 710068, China

Email: lzh_nwpu@hotmail.com Tel.: 029-88488002

ABSTRACT

The high availability issue is an important research topic in distributed file systems. The distributed file systems' resiliency refers to the ability of the important files to tolerate intrusion. The resiliency technology seeks an active strengthening of the systems, rather than protecting its system infrastructure using static defensive measures such as encryption, IDS, and firewalls. System resiliency involves the dynamic use of replication to achieve intrusion tolerance so that even undetected attacks do not cause system failure. The resilient file systems can deliver crucial data the essential properties such as confidentiality, integrity and availability, despite the presence of intrusions. This paper describes the resiliency approach to helping assure that the distributed file system is robust in the presence of attack and will tolerate fault that result in successful intrusions or other disaster failures. Included are discussions of resiliency as an integrated infrastructural framework, the specification of resiliency requirements, strategies for achieving resiliency, and techniques and processes for affording file system resiliency.

Keywords: Distributed file system, Resiliency, Fault tolerance, Reliability, Availability.

1. INTRODUCTION

The widespread availability of networks such as the Internet as a medium for communication has prompted a proliferation of both stationary and mobile devices capable of sharing and accessing data across networks spanning multiple administrative domains. Such collaboration increases user dependence on remote elements of the system which are often out of their control and heightens their need for effective security systems. The shift toward the networking paradigm has resulted in a fundamental re-evaluation of data security. Especially for distributed file system, it is more important to pay attention to the data availability.

Key to the concept of resilience is affording the essential services (and the essential properties that support them) continually within an operational system. Essential services are defined as the functions of the system that must be maintained when the environment is hostile or failures or accidents are detected that threaten the system. There are typically many services that can be temporarily suspended when a system is dealing with an attack or other extraordinary environmental conditions. Such a suspension can help isolate areas affected by an intrusion and free system resources to deal with its effects. The function of resilient system should adapt to preserve essential services.

The capability of a resilient file system is linked to fulfill its mission in a timely manner to its ability to deliver essential

data services in the presence of attack, accident, or failure. Ultimately, mission fulfillment must survive not any portion or component of the system. If an essential data service is lost, it can be replaced by another data that supports mission fulfillment in a different location.

The rest of the paper is structured as follows. Section 2 briefly surveys related work. The specification of resiliency requirements is presented in Section 3. Next, we describe the infrastructural framework of resilient files system in Section 4. Section 5 describes some techniques and processes for affording file system resiliency. Finally, we draw a conclusion in Section 6.

2. RELATED WORK

The application of survivability systems concepts in software design is not new. Jehuda and Israeli [1] proposed a control system for dynamically adapting a software configuration to accommodate varying runtime circumstances impacting on real-time performance. In CHAOS [2], real-time systems are adapted with the use of an entity-relation database modeling system structure. Survivability systems ideas have been used in distributed application management as well. Meta [3] is architecture and a tool that uses a non-hierarchical control system to optimize performance in fault-tolerant distributed systems using Isis. Distributed application management (e.g., [4,5]) employs services supporting the dynamic management of distributed applications. Network management uses survivability concepts to manage networks and their running software [6, 7].

However, the major objective in such work is to monitor and improve application or network performance in traditional dimensions, e.g., runtime efficiency. By contrast, the resilient technology is targeted at enhancing the availability and reliability of distributed file systems.

In system availability aspect, automatic synthesis of self-recovering micro-architectures has been previously addressed. An algorithm that intertwines checkpoint insertion and scheduling to synthesize self-recovering micro-architectures for supporting fault-recovery in hardware was first presented in [8]. Guerra et. al have developed synthesis for built-in self-repair using redundant modules[9]. More recently, Blough, et. al. [10] presented an algorithm for recovery point insertion in recoverable micro-architectures. These RT-level techniques for transient and permanent fault-tolerance have been successful in certain situations. The main target for built-in-self-repair (BISR) techniques for yield enhancement are systems that are bit-, byte-, or digit- sliced, and in particular memories [11] and PLAs [11,12].

* Supported by the National High-Tech Research and Development Plan of China under Grant No. 2003AA142060

3. CHARACTERISTICS OF RESILIENT FILE SYSTEMS

The most important characteristic of resilient file systems is their capability to deliver essential services in the face of attack, failure, or incident.

Central to the delivery of essential services is the capability of a data system to maintain essential properties (i.e., specified levels of integrity, confidentiality, availability and other quality attributes) in the presence of attack, failure, or incident. Thus, it is important to define minimum levels of quality attributes that must be associated with essential services.

This section will now detail how the three security aspects (availability, integrity and confidentiality) can be interpreted in behavioral and protective terms. See Table 1, which describes the situation for resilient file systems security.

Table 1 Resilient file systems security and its aspects

	Key Property	Description	Attribute
Resilient file systems security	Availability	Prevention of the unauthorized withholding of data files	Behavioral (User)
	Integrity	Prevention of the unauthorized modification of data files	Protective (Non-user)
	Confidentiality	Prevention of the unauthorized disclosure of information	Behavioral (Non-user)

Availability

Availability is the ability of the file system to deliver its data service to the authorized user. It is thus a behavioral concept. The authorized users are the users that are the intended receivers of the service that the system delivers. In the following we call the authorized user(s) the User. This may be a human or an object: a person, a computer, a program etc. We have chosen to regard all potential users except the authorized users as unauthorized users. Unauthorized users are called Non-users. Therefore, availability as a security aspect has the same meaning as the availability attribute of file system's resilience.

Integrity

Integrity is the prevention of unauthorized modification or the deletion or destruction of file data assets. Integrity is violated by means of an attack, which is normally performed by a Non-user, but may also be performed by a User who is abusing his/her authority. Thus, integrity is a protective quality of a data system and characterizes the resilient file system's ability to withstand attacks.

Confidentiality

Confidentiality is the ability of the file system to deny the Non-user access to confidential information. It is thus a behavioral concept but, unlike other attributes, it defines system behavior with respect to a Non-user. It actually defines to what extent information should be accessible, or rather not accessible, to Non-users. Therefore, confidentiality is behavioral concept, parallel to reliability, availability and safety. Confidentiality can also be understood in a broader sense, i.e., the prevention of the delivery of data service to the Non-user, even if this service delivery would not include harm to the User or disclosure of secret information. The term exclusivity has been proposed for this broader concept [13].

The characteristics of resilient file systems discussed above leads to a modified understanding of security as two concepts: protective security and behavioral security. Protective security is simply regarded as a form of data fault prevention, namely fault prevention with respect to intentional faults and attacks. Behavioral security is an integrated part of dependability and can not readily be distinguished from it.

In view of this discussion, we arrive at two generic types of behavioral attributes: reliability/availability and confidentiality. Confidentiality relates to the denial-of-service to Non-users, i.e. unauthorized users shall not be able to obtain data information from the file system, nor be able to use it in any other way. Reliability and availability have been merged, since they both refer to delivery-of-service to the User. This does not mean that they are the same. They are merged as they both reflect delivery-of-service to the authorized user, even if different aspects of this delivery. The safety attribute characterizes a certain failure mode of the file system: it denotes the nonoccurrence of catastrophic failures. Note that failures can be of both a "reliability" type, i.e., related to the User, as well as a "confidentiality" type, i.e., related to the Non-user.

4. RESILIENT FILE SYSTEMS ARCHITECTURAL MODEL

Nowadays, computer security has made valuable contributions to the protection and integrity of information systems. However, computer security has traditionally been used as a binary term, which suggests that at any moment in time a system is either safe or compromised. We believe that this use of computer security engenders viewpoints that largely ignore the aspects of recovery from the compromise of a system and aspects of maintaining services during and after an intrusion. Such an approach is inadequate to support necessary improvements in the state of the practice of protecting computer systems from attack. In contrast, the term resilient systems refer to systems whose components collectively accomplish their mission even under attack and despite active intrusions that effectively damage a significant portion of the system.

So a resilient file system must provide trustworthy data to the User and deny service to Non-users. That means the file system resilient technology insures the content of data files dependability and readiness for use. Even if the data is tampered with by vicious intruder, the resilient system can serve the User correctly by the replicas of other locations. Figure 2 depicts an architecture model for resilient file systems.

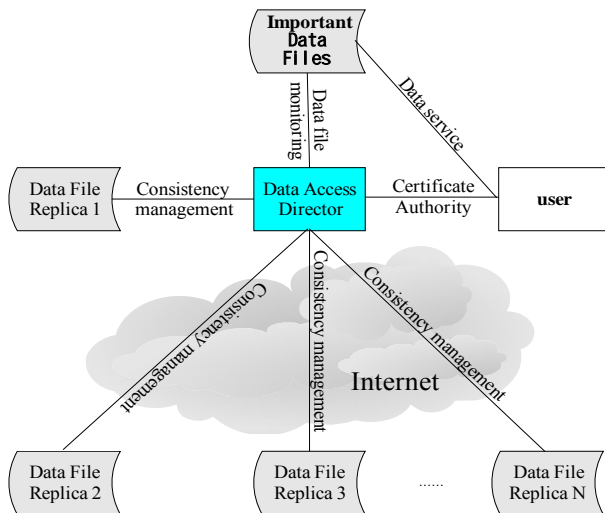


Figure 1 Resilient Distributed File Systems Architectural Model

To tolerate attacks, resilient file systems may choose to statically replicate critical data files, thereby forming replica groups, as shown in Figure 1. The replicas of important data files must be distributed in different hosts which are placed in different networks and different locations. The distributed multi-replicas redundancy technology can avoid data service interrupting even if the file and some of its replicas are tampered. The number of the file replicas is determined by the importance of the data and level of file resilience.

In the resilient distributed file systems architectural model, the centre is the data access director. In the following sections we call the data access director the Dad simply. The Dad monitors the important data files incessantly. If the data files are changed illegally, the Dad recovers them from the replicas. If the data files naturally update, the Dad synchronizes the replicas with the data files. If the User wants to access the data service, they should contact the Dad to get the certificate authority and acquire the access control authorization. Then the User does its business on the protected data files.

Using resiliency, periodic liveness checks are performed by the Dad too. These checks are *not* designed to detect an intrusion attack, but rather seek to determine whether a replica is not performing as expected or not. If a replica is detected as compromised during a liveness check, it will be destroyed and rebuilt using the uncompromised residual members of the replica group. The Dad is accountable for consistency management of all the members in the replica group.

The Dad is the core of the distributed resilient file system. It monitors the protected files timely to insure the integrity, provides data confidentiality by User access control and performs availability by multi-replicas redundancy technology. Its main function can be illuminated by the following program in C pseudocode.

```
Function Data Access Director {
    Data_digest0=Data digest (protected files);
    #create the replicas, n is the number of replicas.
    for ( i = 1; i <= n ; i++ ) {
        Replica[i]=Created replica(protected files);
    }
}
```

```
# the user access control.
user_ authorization= certificate authority (user);
if (the User){
    Get data service(protected files, user_ authorization);
    Data_digest0=Data digest (new state of protected files);
    Replicas state updating ( protected files );
}
else {
    reject data service();
}
#monitoring the protected files timely and replicas
#consistency management;
While(periodic time interval ){
    Data_digest1=Data digest (protected files);
    if (Data_digest1 != Data_digest0){
        Recovery(protected files);
    }
    for ( i = 1; i <= n ; i++ ) {
        Data_digest[i]=Data digest (Replica[i]);
        if (Data_digest[i] != Data_digest0){
            destroy(Replica[i]);
            Replica[i]=Created replica(protected files);
        }
    }
}
```

5. RESILIENT DESIGN AND IMPLEMENTATION STRATEGIES

In order to be resilient for intrusion, resilient file systems must interact with all the secure strategies to strengthen system and collaborate with other systems to safety the internet environment. But the importance of the resilient file systems is to ensure continuity of essential data services. So providing file system with attack tolerance is an important problem. The flexibility that file system can offer makes it a natural choice to implement a significant portion of the attack tolerance of resilient distributed file systems.

As shown in section 4, the architecture model of resilient file systems provides integrity, confidentiality, availability for the protected data files. In this section, we describe the implementation techniques of resiliency design.

5.1. Remote Data Backup Technology

One important thing is the regular, reliable remote backup when it comes to preventive maintenance and information system care. No matter how well you treat your system, no matter how much care you take, you cannot guarantee that your data will be safe if it exists in only one place. Unfortunately, people regularly lose large quantities of data because they haven't backed it up. The remote data backup protects your valuable data against software errors, disk crashes, user error, theft, and virus attacks. The resilient file systems introduce the remote backup technology to decentralize the replicas. Even if the protected data files and some of their replicas are compromised, the User still can get the correct service in time.

Since making a full copy of a large file system can be a time-consuming and expensive process, the resilient file system makes full backups only once, and stores only changes in future. The Dad sends the changes to all the replicas and

rebuilds the new state replicas. These are called "incremental" backups and you don't have to use tape as your backup medium; it is both possible and vastly more efficient to perform incremental backups with high-efficiency network transfer tools. The resilient file systems can use high-efficiency network transfer tool rsync^[14] to achieve remote update of protected file replicas.

The high-efficiency network transfer tool rsync operates on SSH as a secure channel and send/receive only the bytes inside files that have been changed since the last replication. Rsync adopts remote synchronization protocol to transfer only the differences between two sets of files across the network connection, using an efficient checksum-search algorithm.

The rsync algorithm supposes we have two general purpose computers α and β . Computer α has access to a file A and β has access to file B, where A and B are "similar". There is a slow communications link between α and β . The rsync algorithm consists of the following steps:

1. β splits the file B into a series of non-overlapping fixed-sized blocks of size S bytes. The last block may be shorter than S bytes.
2. For each of these blocks β calculates two checksums: a weak "rolling" 32-bit checksum and a strong 128-bit MD4 checksum.
3. β sends these checksums to α .
4. α searches through A to find all blocks of length S bytes (at any offset, not just multiples of S) that have the same weak and strong checksum as one of the blocks of B. This can be done in a single pass very quickly using a special property of the rolling checksum.
5. α sends β a sequence of instructions for constructing a copy of A. Each instruction is either a reference to a block of B, or literal data. Literal data is sent only for those sections of A which did not match any of the blocks of B.

The end result is that β gets a copy of A, but only the pieces of A that are not found in B (plus a small amount of data for checksums and block indexes) are sent over the link. The algorithm also only requires one round trip, which minimizes the impact of the link latency.

5.2. Replica Consistency Management

The most common use for replica consistency management is checking data file replicas for corruption. A message digest or checksum calculation might be performed on the protected data files. Making the same calculation of all its replicas and comparing the results, you can determine whether some of the replicas are corrupted or not. If the results match, then the replica is likely accurate.

A message digest is a compact digital signature for an arbitrarily long stream of binary data. An ideal message digest algorithm would never generate the same signature for two different sets of inputs. We can use the message digest to ensure that all the replicas have the same content or status with the protected files.

The Dad can use MD5 algorithm or SHA-1 algorithm to calculate the message digest of the protected data file and all its replicas. When a data file begins to be protected by the Dad, the Dad calculates its message digest and builds its replica group. The message digest will be used to maintain consistency of the replicas and be renewed when the data file

is changed normally. The Dad checks the message digest of each replica periodically. If one of replicas' message digest doesn't match with the protected files, the Dad will rebuild the replica. If protected file is changed by normal operations, the Dad will update all the replicas in the group.

5.3. Recovery Technology

The Dad monitors the protected files activity and responses to its changes timely. If the protected file is changed by the operations with permission of the Dad, the protected file keeps the changes and updates all its replicas; if the protected file is tampered with by vicious intruder, the Dad recovers the protected file from the replicas.

In the distributed resilient file system, the recovery can be looked on as converse process of one backup process. When the protected file is changed and needs recovery, the Dad verifies the reliability of all the replicas, and then chooses a replica host whose cost of transfer of data to the protected file host is minimal as the backup source to implement the recovery process of the destined protected file with the remote data transfer technology discussed in section 5.1.

Requirements for recoverability are what most clearly distinguish resilient systems from other systems that are merely secure. The recovery technology of distributed resilient system contributes to the system's ability to maintain essential services during intrusion and ensures the business continuity of data service.

5.4. Service Access Control

The resilient distributed file system offers several security advantages not available in ordinary distributed file system. One advantage is access control. You can restrict who has access to data on your computer, or on the network using access control implemented by the Dad. The Dad sets permissions to define the type of access granted to a user or group. For example, it can grant Read and Write permissions to the User for some data files. When the Dad set up permissions, it can specify the level of access for groups and users. For example, it can let one User read the contents of a file, let another User make changes to the file, and prevent the Non-user from accessing the file.

The service access control provides confidentiality guarantee to the distributed resilient file systems. It is an important task in the data security technology.

6. CONCLUSION

This paper has described the notion of file systemic resiliency and discussed the implementation issues. It shows how the concepts can be applied in the distributed file system to tolerate intrusion. The file system resilient technology adopts the active data security strategy to provide data service continuity and introduces techniques of system survivability to ensure availability, integrity and confidentiality of the important files. In future, there will be lots of work to be done in the research of resilient file systems.

7. REFERENCES

- [1] J. Jehuda and A. Israeli, "Automated Meta-Control for

- Adaptive Real-Time Software”, *Real-Time Systems*, Vol. 14, 1998, pp. 107-134.
- [2] P. Gopinath, R. Ramnath, and K. Schwan, “Database Design for Real-Time Adaptations”, *Journal of Systems and Software*, Vol. 17, 1992, pp. 155-167.
 - [3] K. Marzullo, R. Cooper, M. D. Wood, and K. P. Birman, “Tools for Distributed Application Management”, *IEEE Computer*, August 1991, pp. 42-51.
 - [4] M. A. Bauer, R. B. Bunt, A. El Rayess, P. J. Finnigan, T. Kunz, H. L. Lutfiyya, A. D. Marshall, P. Martin, G. M. Oster, W. Powley, J. Rolia, D. Taylor, and M. Woodside, “Services Supporting Management of Distributed Applications and Systems”, *IBM Systems Journal*, Vol. 36 No. 4, 1997, pp. 508- 526.
 - [5] Tivoli Systems, “Tivoli and Application Management”, White paper, http://www.tivoli.com/o_products/html/body_map_wp.html, 1998.
 - [6] B. Boardman, “Network Management Solutions Lack Clear Leader”, *Network Computing*, August 15, 1998, pp. 54-67.
 - [7] Computer Associates, “Enterprise Management Strategy: Managing the New Enterprise”, White paper, <http://www.cai.com/products/unicent/whitepap.html>, 1996.
 - [8] A. Orailo~lu and R. Karri. "Coactive Scheduling and Checkpoint Determination during the High Level Synthesis of Self Recovering Microarchitectures," *IEEE Trans on VLSI Systems*, 1994:2(3): 304-311.
 - [9] L.M. Guerra, et al., "High Level Synthesis Techniques for Efficient Built-in Self Repair", *IEEE Workshop on DFT in VLSI systems*, 1993: pp. 41-48.
 - [10] D. M. Blough, F. J. Kurdahi, and S. Y. Ohm, "Optimal Recovery Point Insertion For High Level Synthesis of Recoverable Microarchitectures," *FTCS*, 1995.
 - [11] Daniel P. Siewiorek , Robert S. Swarz, *Reliable computer systems (2nd ed.): design and evaluation*, Digital Press, Newton, MA, 1992
 - [12] I. Koren, D.K. Pradhan, "Introducing Redundancy into VLSI Designs for Yield and Performance Enhancement", *FTCS 15*, 1985:pp. 330-335.
 - [13] D. E. Denning, “A New Paradigm for Trusted Systems”, *Proceedings of the IEEE New Paradigms Workshop*, 1993: pp. 36-41.
 - [14] A. Tridgell and P. Mackeras. The rsync algorithm. Technical Report, http://samba.anu.edu.au/rsync/tech_report/tech_report.html, Australian National University, 1998.



LI Zhonghua was born in 1976. He is a Ph.D. candidate in School of Computer Science, Northwestern Polytechnical University. He received his BS and MS degrees from Northwestern Polytechnical University in 1999 and 2002 respectively. His research interest includes fault-tolerant distributed systems, parallel processing, network security, and intelligence decision

Supporting System.

Efficient Scheduling of Task Graphs to Multiprocessors Using a Simulated Annealing Algorithm

Wenbo Xu, and Jun Sun

School of Information Technology, Southern Yangtze University, Wuxi, Jiangsu, 214036, People's Rep. of China
Email: xwb@sytu.edu.cn; sunjun21c@163.com

ABSTRACT

Given a parallel program modeled by a directed graph (DAG), the problem of scheduling the tasks of the program to multiprocessors has been proven to be NP-complete. For this reason, heuristics are usually used to tackle the scheduling problem. But the existing heuristic methods have many disadvantages, such as high time complexity, lack of scalability and no performance guarantee with respect to optimal solutions. To overcome or weaken these defects, in this paper, we employ a stochastic search technique in list scheduling and propose a simulated annealing algorithm for task scheduling (SATS). We also devise a novel topological sorting algorithm, stochastic topological sorting algorithm (STS), to generate an initial scheduling list. Our experiment results show that by setting control parameters properly, the proposed performs better than other heuristic algorithms, with affordable running time.

1. INTRODUCTION

On message-passing multiprocessor systems, the goal of scheduling tasks of a parallel program, which modeled by an acyclic graph (DAG), is to minimize the completion time of the program. This multiprocessor scheduling problem is NP-complete even simplifying assumptions and becomes more complex under relaxed assumptions such as arbitrary precedence constraints, and arbitrary task execution and communication times. Due to its intractability, many polynomial-time heuristics are employed to tackle the scheduling problem. The rationale of these heuristic methods is to sacrifice optimality for the sake of reduced time complexity. One of the important classes of heuristic algorithms is list scheduling, which is usually used for a bounded number of processors. (e.g., MCP[4], DPS[8]). It has been shown that list-scheduling algorithms perform well at a relatively low cost compared to other high-cost scheduling algorithms for bounded number of processors. In list scheduling, two approaches can be distinguished. One approach, on which we exert emphasis in this paper, is list scheduling with static priorities (LSSP), the other with dynamic priorities (LSDP).

In LSSP, the tasks are scheduled in the order of their previously computed priorities on the task's "best" processor. Thus, at each scheduling step, first the task is selected and afterwards its destination processor. Usually, if the performance is the main concern, the "best" processor is considered the processor enabling the earliest start time for the given task, whereas if the speed is given the first rank, the selected processor is the processor becoming idle the earliest when the task is scheduled.

Even though the heuristics including the list scheduling are shown to be effective in general, they generally cannot generate optimal solutions, and there is no guarantee of their performance as well as their scalability with the problem size in general. In view of the drawbacks of the existing list scheduling

heuristics, we commit ourselves to framing a new scheduling scheme, which has a high capability to generate optimal solution and is also fast and scalable. To obtain high quality solutions and achieve a reduced time complexity, we devise a simulated annealing formulation of the scheduling problem in which scheduling lists are systematically combined by using a so-called stochastic topological sorting algorithm and the stochastic research to determine an optimal scheduling list.

The rest of the paper is organized as follows. In the next section we provide the problem statement. In Section 3 we give a brief survey of simulated annealing techniques. In Section 4 we present the proposed simulated annealing scheduling algorithm including the stochastic topological sorting algorithm. In Section 5 we describe our experiment study and its results. Finally, we provide some concluding remarks.

2. PROBLEM STATEMENT

A parallel program can be modeled by a directed acyclic graph (DAG) $G = (V, E)$, where V is a set of v nodes, representing the tasks, and E is a set of e directed edges, representing the communication message. Edges in a DAG are directed and, thus, determine the precedence constraints among the tasks. The cost of node n_i , denoted as $w(n_i)$, represents the computation cost of the node. The weight of an edge (n_i, n_j) , denoted by $c(n_i, n_j)$, represents the communication cost of the message. The communication-to-computation-ratio (CCR) of a parallel program is defined as its average communication cost divided by its average computation cost on a given system.

The source node of an edge is called a *parent* node, while the destination node is called a *child* node. A node with no parent is called an *entry* node and a node with no child is called an *exit* node. The precedence constraints of a DAG dictate that a node can only start execution after it has gathered all of the messages from its parent nodes. The communication cost between two nodes allocated to the same processor is assumed to be zero. Therefore, the data available time (DAT) of a node depends heavily on the processor to which the node is scheduled. If n_i is scheduled, $ST(n_i)$ and $FT(n_i)$ denote the start time and finish time of n_i , respectively. After all nodes have been scheduled, the Schedule Length is defined as $\max_i \{FT(n_i)\}$ across all nodes. The objective of scheduling is to minimize the schedule length by proper allocation of the nodes to the processors and arrangement of sequencing of the nodes without violating the precedence constraints. An example of DAG shown in Fig. 1, will be used as an example in the subsequent discussion.

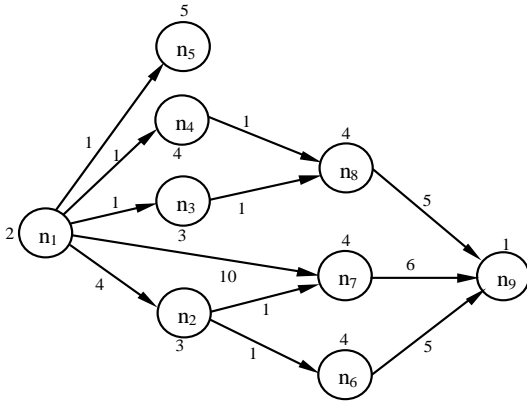


Fig. 1. A DAG example

It is an example of a task graph. There are 9 task nodes in the DAG with each node having its own computation cost. Every directed edge represents a precedence constraint between two nodes and the weight of the edge represents the communication cost

The target system is also assumed to be a fully connected distributed memory multiprocessor system with no regard to link contention and scheduling of message. An example processor graph is shown in Fig. 2.

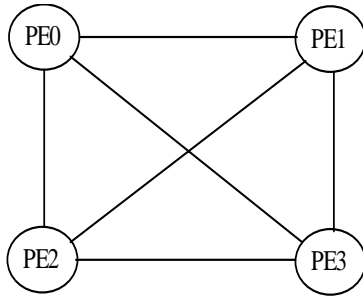


Fig. 2. Processor graph

The figure shows a graph of a 4-processor fully connected target system. The processing elements (PEs) in the target system may be heterogeneous or homogeneous. We assume the communication links are homogenous

3. A BRIEF SURVEY OF SIMULATED ANNEALING ALGORITHM

Simulated Annealing (SA) algorithms approach optimization problems by randomly generating a candidate solution and, then, making successive random modifications. A temperature parameter is used to control the acceptance of modifications. Initially, the temperature is at a high value and it is decreased over time. If the modified solution is found to have a better fitness than its predecessor, then it is retained and the previous solution is discarded. If the modified solution is found to be less fit than its predecessor, it is still retained with a probability directly related to the current temperature. As execution of the algorithm continues and the temperature becomes cooler, it becomes less likely that unfavorable solutions are accepted. By using this approach, it is possible for an SA algorithm to move

out of local minima early in execution, and more likely that good solutions will not be discarded late in the algorithm's execution.

Although SA algorithms are conceptually simple, finding optimal parameter for SA, i.e., initial temperature, α (a constant value between 0.0 to 1.0 that is multiplied to the current temperature at fixed intervals to obtain new temperature values), etc., is by no means simple for straightforward. First of all, setting parameters for SA is problem dependent, and it is best accomplished through trial and error. Furthermore, previous studies with SA have demonstrated that SA algorithms are very sensitive to parameters, and their performances are largely dependent on fine-tuning of the parameters. The problem dependent nature of setting parameters for SA and SA algorithms' sensitivity to parameters limit the effectiveness and robustness of SA algorithms.

4. THE PROPOSED ALGORITHM

As mentioned in Section 1, the simulated annealing technique, in our proposed algorithm, is used to obtain a scheduling list of nodes whose schedule length through afterwards task allocation is minimum. Thus, a scheduling list complying with the precedence constraints among task nodes is a candidate solution to the scheduling problem, and the task allocation procedure is the very goal function of the problem. In the following text, we'll set forth the designing methodology of the algorithm.

4.1 The Task Allocation Strategy

The task allocation strategy in LSSP is dependent on the determination of the "best" processor to which a task node is to be scheduled. According to the statement in Section 1, as we emphasize on the performance of the algorithm, the "best" processor is the processor enabling the earliest start time for the given task n_i . In order to select such a processor, the earliest start time of n_i on each processor p_j must be ascertained first by the following approach.

Computation of $ST(n_i, P)$ If m modes $\{n_{p_1}, n_{p_2}, \dots, n_{p_m}\}$ have been scheduled on processor $P (m \geq 0)$. The earliest start time of n_i on processor P can computed as follows.

$$ST(n_i, P) = \max\{FT(n_{p_m}, P), DAT(n_i, P)\} \quad (1)$$

where $FT(n_{p_m}, P)$ is the finish time of n_{p_m} , and $DAT(n_i, P)$ is the data available time of n_i if scheduled on processor P .

Based on the "best" processor policy adopted in our algorithm and the computation of $ST(n_i, P)$, the task allocation strategy, scilicet, the goal function of the scheduling problem is referred to as the *start-time minimization* procedure, which is outline as follows.

Start-Time Minimization

$$(1) \quad \forall j, \text{RedayTime}(P_j)=0;$$

```

(2) while the scheduling list  $L$  is not empty do
(3)   remove the first node  $n_i$  from  $L$ ;
(4)    $\text{Min\_ST} = \infty$ ;
(5)   for  $j=0$  to  $p-1$  do
(6)      $\text{This\_ST} = \max\{\text{ReadyTime}(P_j), \text{DAT}(n_i, P_j)\}$ ;
(7)     if  $\text{This\_ST} < \text{Min\_ST}$  then  $\text{Min\_ST} = \text{This\_ST}$ ;
        $\text{Candidate} = P_j$ ; endif
(8)   endfor
(9)   schedule  $n_i$  to  $\text{Candidate}$ ;
        $\text{ReadyTime}(\text{Candidate}) = \text{Min\_ST} + w(n_i)$ ;
(10) endwhile

```

(In the procedure above, p is the number of processors in the target system, and the ReadyTime of a processor is the finish time of the task scheduled on the processor the latest.)

4.2 Stochastic Topological Sorting Algorithm

Having determined the task allocation strategy, we are to find out how to generate an initial scheduling list, namely an initial solution. Since topological order of the DAG is a valid scheduling list that complies with the precedence constraints among the task nodes, the topological sorting algorithm (TS) can serve the purpose.

But the TS has two fatal disadvantages, one of which is that it is based on the depth-first search so that the topological orders generated by the TS algorithm cannot cover the set of the feasible solutions for the problem, the other of which the topological order is fixed because it is subject to the storage structure of the DAG in the computer. In order to overcome these defects of the TS algorithm, we devise a novel sorting algorithm, a stochastic topological sorting algorithm (STS).

The STS algorithm, which described below, along with our adopted terminating criteria, is able to reduce the search time of the proposed algorithm considerably.

Stochastic Topological Sorting Algorithm (STS)

```

(1)   Initialize a linked list of zero-in-degree nodes (ZL);
(2)   Initialize a scheduling list SL;
(3)   for  $i=1$  to  $v$  do
(4)     compute the in degree of  $n_i$ ,  $\text{in\_degree}(n_i)$ ;
(5)     if  $\text{in\_degree}(n_i) == 0$  then insert  $n_i$  into ZL;
       endif
(6)   endfor
(7)   while the ZL is not empty do
(8)     select randomly a node  $n_x$  from ZL and put it into
       SL;
(9)     delete the node from ZL;
(10)  for each child node of  $n_x$ ,  $n_y$  do
(11)     $\text{in\_degree}(n_y) = \text{in\_degree}(n_y) - 1$ ;
(12)    if  $\text{in\_degree}(n_y) == 0$  then insert  $n_y$  into ZL;
       endif
(13)  endfor
(14) endwhile
(15) Output the SL;

```

Take the DAG in Fig. 1 for an illustrating example. If the

topological order $(n_1, n_2, n_7, n_4, n_3, n_8, n_6, n_9, n_5)$

is generated, the steps, as shown in Table 1, will be executed by the STS algorithm. The fixed TS, however, cannot yield such a scheduling list at all.

Table 1. The steps executed by the STS algorithm

The Zero-in-degree Linked List of the nodes	Scheduling List
n_1	n_1
n_2, n_3, n_4, n_5	n_2
n_3, n_4, n_5, n_6, n_7	n_7
n_3, n_4, n_5, n_6	n_4
n_3, n_5, n_6	n_3
n_5, n_6, n_8	n_8
n_5, n_6	n_6
n_5, n_9	n_9
n_5	n_5

Nine steps must be executed by STS algorithm to combine the scheduling list above. The content of ZL and the randomly selected node in each executing step is shown

4.3 Modification of the Scheduling List

In a SA algorithm, the modification of a solution that prescribes the neighborhood of the solution is to perturb the solution to generate a novel one. In the task-scheduling problem, a topological order can be transformed into another topological order by swapping two adjacent nodes in the list without violating the precedence constraints. The two nodes are interchangeable if they are not lying on the same path in the DAG. For example, the scheduling list $(n_1, n_4, n_2, n_3, n_7, n_6, n_8, n_5, n_9)$ can be modified into the scheduling list $(n_1, n_2, n_4, n_3, n_7, n_6, n_8, n_5, n_9)$

by swapping n_4 and n_2 . Therefore, before the swap operation is done, we must check whether the two randomly selected adjacent nodes are interchangeable. We define the modification of a given scheduling list as a swap of two interchangeable randomly selected adjacent nodes.

4.4 The Temperature Parameter and Terminating Criteria

Because of the SA's sensitivity to the parameters, the temperature parameter and terminating criteria are worth taking pain to study. In our recent experiment, we adopt these parameters as follows.

Initial Temperature t_0 To get the initial temperature t_0 , we modify, in our algorithm, a given scheduling list for some times, count the times the goal function (schedule length) augments, and thus compute the average value of the increment of goal function, denoted as $\overline{\Delta f^+}$. Therefore the initial temperature can be obtained by

$$\chi_0 = \exp\left(-\frac{\overline{\Delta f^+}}{t_0}\right) \quad (2)$$

that is,

$$t_0 = \frac{\overline{\Delta f^+}}{-\ln \chi_0} \quad (3)$$

where χ_0 denotes the initial rate acceptance and is set to be 0.8 in our experiment.

Cooling Function. The cooling function is what controls the attenuation of the temperature parameter during the stochastic search. The formulation below is the cool function we adopted in our proposed algorithm.

$$t_{k+1} = \alpha \times t_k \quad (4)$$

The Length of Markov Chain. The length of Markov chain is determined as $L = v$, where v is the number of the nodes in the DAG, representing the problem size.

Terminating Criteria. In our algorithm, if the solution is not change for s successive Markov chains, the execution of the algorithm terminates. The value of s is set to be v in the experiment.

4.5 SA Algorithm for Task Scheduling Problem

Having determined some important factors in SA, we describe our proposed algorithm below.

Simulated Annealing Algorithm for Task Scheduling Problem (SATS)

- (1) Generate a initial scheduling list sl_0 by STS;
- (2) Schedule the sl_0 by start-time minimization step and get the schedule length SL_0 ;
- (3) Compute the initial temperature t_0 and let $t = t_0$;
- (4) Let $L = v$, $sl = sl_0$, $f = SL_0$;
- (5) **while** the terminating criteria is not satisfied **do**
- (6) **for** $l = 0$ to L **do**
- (7) modify the sl to generate sl_1 ;
- (8) schedule sl_1 by start-time minimization step and let
- (9) $f_1 = SL_1$;
- (10) $f = f_1 - f$;
- (11) **if** $\exp\left(-\frac{\Delta f}{t}\right) > \text{random}(0,1)$ **then** $sl = sl_1$;
- (12) **endif**
- (13) **endfor**
- (14) $t = \alpha * t$;
- (15) **endwhile**

5. EXPERIMENT RESULTS

To test the performance of the SATS algorithm, we use a group of randomly selected task graphs to perform SATS, DCP, and DSC algorithms respectively. Afterwards, the average schedule lengths of these graphs resulted from these algorithms are compared.

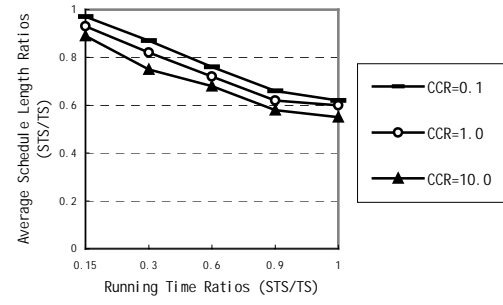


Fig. 3. Performance comparison among three algorithms

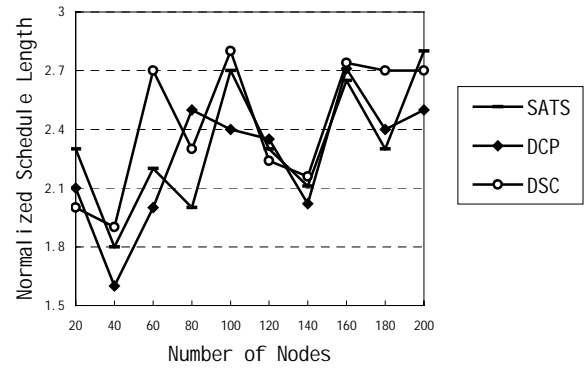


Fig. 4. Running time ratios of SATS' using STS and TS

The figure 3 shows the performance comparison among three algorithms. We also test the speed of SATS algorithm by comparing the running time SA algorithms using fixed topological sorting and stochastic topological sorting respectively.

6. CONCLUSIONS

In the above text, we have presented a simulated annealing algorithm for task scheduling on multiprocessors. The major motivation of using a simulated annealing technique is that SA algorithms can theoretically find out by stochastic search an optimal solution to the problem with a tolerant time complexity. However, due to the NP-completeness of the scheduling problem, in practice, we use the SA technique in selecting an optimal scheduling list, and meanwhile, adopt a heuristic idea in the task allocation procedure. The purpose of doing so is to arrive at a tradeoff between the performance and the time complexity of the algorithm.

One noteworthy thing is the sensitivity of the SA algorithm to the temperature parameters. Our current job concentrates on seeking more subtle parameters leading to better performance and low time complexity for the algorithm.

7. REFERENCES

- [1] Ishfaq Ahmad, Yu-Kwong Kowk: On Parallelizing the Multiprocessor Scheduling Problem. IEEE Transaction on Parallel and Distributed Systems, Vol. 10, No. 4 (1999)

- 414-431
- [2] Sekhar Darbha, Dharma P. Agrawa: A Task Duplication Based Scalable Scheduling Algorithm for Distributed Memory Systems. *Journal of Parallel and Distributed Computing* 46, (1997) 15-27
 - [3] Andrei Radulescu, Arjan J.C. van Gemund: Low-Cost Task Scheduling for Distributed Memory Machines. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 13, No. 6 (2002) 648-658
 - [4] Min-You Wu, Daniel D. Gajski: Hypertool: A Programming Aid for Message-Passing Systems. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 1, No. 3 (1990) 330-343
 - [5] Yu-Kwong Kwok: Benchmarking and Comparison of the Task Graph Scheduling Algorithms. *Journal of Parallel and Distributed Computing* 59, (1999) 381-422
 - [6] Tao Yang, Apostolos Gerasoulis: DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 5, No. 9 (1994) 951-967
 - [7] Yu-Kwong Kwok, Ishfaq Ahmad: Dynamic Critical-Path Scheduling: An Effective Technique for Allocating Task Graphs to Multiprocessors. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 5 (1996) 506-521
 - [8] G.-L. Park, B. Shirazi, J. Marquis, H. Choo: Decisive Path Scheduling: A New List Scheduling Method. *Proc. Int'l Conf. Parallel Processing (ICPP)*, Aug. (1997) 472-480
 - [9] M. A. Palis, J.-C. Liou, D.S.L. Wei: Task Clustering and Scheduling for Distributed Memory Parallel Architectures. *IEEE Transactions on Parallel and Distributed Systems*, Vol. 7, No. 1 (1996) 46-55
 - [10] M. R. Garey, D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co. (1979)

Priority Assignment Strategy of Multiple Priority Queues

Zhang Jianhua, Cong Yue

School of Information and Science & Engineering, Shenyang University of Technology,

Shenyang, Liaoning Province, 110023, P. R. China

E-mail: zhangjianhua2002@hotmail.com

ABSTRACT

In this paper, the author analyses and presents that the key of the further development of the applied technology on Internet is parallel processing. The distributed service system which is constructed in the form of client/server, when many users all need to occupy certain resource and nearly send out request messages at the same time, in order to improve the efficiency of the system successively, the service of host may arrange for the handling order according to a certain principle, that is the Priority Assignment Strategy for Multiple Priority Queues based on fuzzy algorithm.

Keywords: parallel programming, fuzzy algorithm.

1. INTRODUCTION

In recent years, with the rapid development of computer and network technologies, the multimedia technology has come into the climax of research and development. Internet has offered a very large scale of information service for many users. The rapid development of Internet establishes the leading position of the TCP/IP protocol suite in the computer network. The development and content expansion of the TCP/IP protocol suite itself (such as, the sustenance of the universal IPv4, the developing Ipv6 possessing 128 bits address mask, address resolution, dynamic host configuration address, digital certificates and data encryption technologies), sustains more powerfully and abundantly the network communication of various multimedia information having different characters. Nowadays, the development tendency is: whether it is the remote system or local area network, whether it is the WAN, Ethernet, ATM, CATV, cell-based wireless, satellite communication; the users, interface of any type of them will be IP protocol. The distributed service, system, constructed in the client/server model, has been one of the main ones on the Internet service. Accordingly, there is a problem appeared: when many users all need to occupy certain resource and nearly send out request messages at the same time, what precedence order should be used to deal with these requests.

These requests may roughly divide into two kinds: the first kind is that data quanta are relatively small and data transferring has the precision of high degree, such as file data or database data; the second one is that data quanta are relatively large and users pay more attention to the data transferring timely and interconnection, but they may relatively allow the data delay for keeping the precision to some extent, such as the data in the VoD system. The second kind of request is related to the research field of the media streaming technology. This article will discuss the first kind of request above.

This kind of request can be divided into two kinds: the request of users for the same contents (the service type that the servers

provide is the same, the contents of the service is also the same; for example, the dispatch of the same resource of a certain type), and the request for the different contents (needing the servers to deal with the different services, or having different requirements in the same service; such as, the dispatch for different resources). In the following parts, will be introduced the solving methods and handling of programming, respectively.

2. COMMUNICATION INTERFACE AND PARALLEL PROCESS

The various services based on TCP/IP protocol suite (for example, the application of HTTP, FTP, etc.) are the data transferring method on the basis of point-to-point. The programming of socket constitutes the Communication Programming Interface between different procedures of a single host and the whole network. One of the interfaces is Windows Socket (WinSock) applied for communication universally and sustains many kinds of protocols on Internet. Moreover, after the continuous development, improvement and sustenance under the Intel, Microsoft, Sun, SGI, Informix, Novell and other companies, it has become the standard of windows network programming in fact. First of all, the client end tries to connect to some server on the network, after the success of connecting, it will send out requests to the server. In this course, the network communication IP address and the number of transmission port have already decided which server should be used to offer the service and the service type on the network.

As for the service order, there are three kinds of methods, FIFO queuing, the demand priority queuing and the priority according as the size of the task.

Considering how to realize the three principles above reasonably, here we present the Priority Assignment Strategy for Multiple Priority Queues based on fuzzy algorithm and a doubly linked list deal with the waiting queue to improve the system ability.

3. ALGORITHM DESIGN

System Model

When many clients are sending out their requests almost at the same time, the amount of information may be different and the delay time allowed for waiting may be also different. Under the premise that the different requirements of different clients are satisfied basically, the service of host may arrange for the handling order successively according to certain principle in order to improve the efficiency of the system. That is, we should grade them roughly at first.

Supposing there are n tasks ($J_i | i=1, 2, \dots, n$) at present,

the responding time of every task is different ($T_i \mid i=1,2,\dots,n$), and there are m priority queues ($P_j \mid j=1,2,\dots,m$), we can set up models according to the following principle.

For every task, to judge that there are two factors of their levels, the responding time and the priority queues, then we have factor sets is $U=\{u_1, u_2\}$, the weighed distribution is

$$A=(a_1, a_2) \quad F(U).$$

They are judged of the factor

$$V=\{\text{the most important, important, ordinary}\}.$$

And the synthesis factor is

$$B=(b_1, b_2, b_3) \quad F(v).$$

There are some fuzzy reflection relations according as following formula

$$f: U \rightarrow F(V)$$

$$U_i \mid f(u_i)=(r_{i1}, r_{i2}, \dots, r_{im}) \quad F(v)$$

There are fuzzy relations

$$R(f)=[f(u_1), f(u_2), f(u_3)]^{-1} \quad m_n \times m$$

Then have the fuzzy conversion

$$T_R: F(u) \rightarrow F(v)$$

$$A \mid T_R(A)=A \circ R$$

Thus the model constructed by (U, V, R) is that formula

$$B=A \circ R=(b_1, b_2, b_3) \quad F(v) \quad (1)$$

That is grade of the task.

Algorithm Analysis

In order to running program quickly and improve system effectually, we may build a table according as formula (1) at first. That is a training process. First we give a set of fuzzy value for the input request messages from client. And then running program and adjust the values, repeated again and again in the process, don't break until we get a set of results conveniently. The data sets form a table for formula (1) and we hold it at server in the host computer.

Of course, each service of host will have own corresponding table, because every service has own transmission port.

Algorithm step 1: When server received a request from any client, it put forward a value from the table according as preparative result quickly. We change the value into a corresponding data of a set that is the flag of priority assignment strategy.

Algorithm step 2: According the priority, insert the task into the waiting-queue.

Algorithm step 3: As we say, the data structure of the waiting-queue is a doubly linked list and the head node of the list will be process at first. The process is an ordinal queue on the list. After processed, the flag of the task will be deleted and the storage will be free. In this way, the compicacy of the step is rely on the period of running time,

unless delete a node in the waiting queue for a accident, such as, the client request be break or hardware error.

4. PROGRAMMING

Technically speaking, a task be divided into several process be operated. CPU parallel processes the multi-process and we name the operation as Context-switching between them. In fact, a process be divide in several thread be operated, the thread is bass unit be operated by CPU.

Multithread programming is very important to the parallel programming. In programming, we must pay attention to the different of operators, such as start, stop, suspend, steep, resume, suspend, join, abort. At the same time, we must consider the operator whether synchronous operator or asynchronous operator.

This is one of process programs which is a C# programming language as following sentences:

```
class IntNode{
public int info; using System;
unsafe public IntNode *next;
unsafe public IntNode(int el, IntNode *ptr) {
    info=el;
    next=ptr; }
}
class IntSLList {
unsafe private IntNode *head;
unsafe private IntNode *tail;
unsafe public IntSLList() {
    head=tail=0; }
unsafe void addToTail(int el) {
    if(tail!=0)
        tail->next=new IntNode(el);
    tail=tail->next; }
unsafe void deleteFromHead() {
    int el=head->info;
    IntNode *tmp=head;
    if(head==tail)
        head=tail=0;
    else head=head->next;
    delete tmp;
    return el; }
unsafe bool isInList(int el) {
    IntNode *tmp;
    for(tmp=head; tmp!=0 && (tmp->info==el); tmp=tmp->next)
        return tmp!=0; }
}
```

5. CONCLUSION

The name Programming interface is concerned with the Programming Communication Interface. Certainly, first the relation is the type of data communication facility, and then, that is an amount of hardware and software is required within each attached computer to handle the appropriate network-dependent protocols. Since in many applications the communicating computers may be of different types, and they may use different programming languages, and more importantly, different forms of data representation interface between user programs. So, we can say, PCI is PPCI (Parallel Programming Communication Interface).

No doubt, parallel processing is of importance for fuzzy algorithm, and any program that will successfully simulate even a small part of fuzzy problem will be complicated. The soft computing is a sub-area of AI, which offers a new methodology using the ideas and methods of computation. The soft computing will be successfully put into the communication field on Internet.

6. REFERENCES

- [1] Barry Wilkinson, Michael Allen, Parallel Programming. China Machine Press, Beijing, 2002
- [2] Fang Shucheng, Wang Dingwei, Fuzzy Mathematics and Optimization *Algorithm*, Science Press, Beijing, 1997
- [3] David E.Culler, Jaswinder Pal Singh, Anoop Gupta. Parallel Computing and System Structure. China Machine Press, Beijing, 2002
- [4] Lin Feng, Zhou Zhengli, Design of Transfer Net System Based on Multi-thread of C/S Model, Computer Engineering and Application, June 2003, pp.184-186
- [5] Zhang Ying, Liu Yangiu, software computing, China Science Press, Beijing, 2002
- [6] Zhu Jing, Fuzzy Control Principle and Application, China Machine Press, Beijing, 1995



Zhang Jianhua is a professor and a head of Interface and Control Lab., School of Information & Science and Engineering, Shenyang University of Technology, P. R. China.



Cong Yue is studying in school of Electrical Engineering, Shenyang University of Technology, P. R. China.

Design and Implement to Load Balancing to the Application Server Cluster

Tang Wei

Math and Computer Science Department, Jiangnan University

Wuhan, Hubei 430056, China

Email: tangwei@jhu.edu.cn Tel: 027-84226927

ABSTRACT

For implementing of the load balancing in the Application server cluster to respond client requests in the 3-tier structure, the lead pipe is essential from Web server to Application server cluster. Based on the Session responding mechanism to the client's requests, we put a method to implement the load balancing in Application server cluster. We designed a Dispatcher module. Dispatcher is offered by the application servers and circulated on the Web server.

Keywords: Load balancing, Server cluster, Session, Distributed compute.

1. THE 3-TIER WEB STRUCTURE AND THE SESSION ACCESS

In the 3-tier structure, the Application server cluster is specialized for the high load in Web business. For typical model of the 3-tier structure, client (request information), procedure (request process) and data (be operated) are separated physically. The Application server cluster is located the middle layer, and circulate on between the browsers and the data resources. As an example, the user input an order form from client and the Web server sends the request of the form process to an application server. Then the application server executes the procedure logics for the request, and the client data is updated. The 3-tier structure has the better transplantation, and can be worked across platforms of the different type. For the high load of request in Web browsers, it is essential to balance the load of the server.

Logically, there is a dispatcher that runs between the Web server and the Application server cluster. By the dispatcher, the business request from browsers is sent to the most vacant application server. Then the browser communicates with the server directly. So, the load balancing in the Application server cluster is realized.

The browser communicates with the Web server by HTTP. When a client sends a web page request, the Web server responds simply. After that it closes the link to the client. Accounting for the un-recollection in HTTP, the Web server send back the cookie to the browser in the responding procedure and keep the cookie on the hard disk of the browser. For example, the IE browser keeps the cookie information in the catalogue: "C:\windows\cookies". When the client visits again, the server requests the browser to check and return the cookie information that has send, so the server can identify the client.

The Session is introduced by the Active Server Pages. A Session begins at a client to input a Web address, and is closed that the client leaves the Web. The main difference between the Session and the cookie is that the cookie is kept in the client, but the Session is kept in the server. The Session is realized of

by the cookies. Because the Cookies inside the client have the client ID, when the browser wants to visit an application server, the Session ID is carried to. So the server is able to identify the client source. The concrete process is following.

When there is a client visit, immediately start the client's Session. As long as the browser support the cookie, a Session ID will be written and put into the cookie. This ID is introduced randomly and is encrypted by MD5. Here the cookie is called the Session's cookie, it can't be written the hard dish of the client. When a Session period end, that cookie is terminated. If the browser does not support cookie, the Session ID will be put into the RUL.

2. DESIGN DISPATCHER

Based on the Session visit mechanism, we can design a module called Dispatcher. Dispatcher is responsible for the Web server to arrive the application server and implementation of the load balancing in the Application server cluster. Dispatcher is provided by the Application server and is circulated on the Web server. The basic design way is following.

When a browser visits the Web server, it's business request will be processed by Dispatcher. According on the cookie, Dispatcher will determinant the request is at the first time or not. If the request is, Dispatcher will interpret the configure document. According to the the load balanced strategy, Dispatcher will distribute the request to an application server which's load is easier, and then the browser Session is allotmented to the application server and write the Session ID into the cookie. At the moment, the browser IP and post address (IP+PORT) is encoded to B64 code. The code is written to the cookie. If the Session that browser request is not the first time, and its right is not over, Dispatcher will gets the B64 code to match the configure document, so that the application server which had responded the browser originally will be determinanted, and the current request be reserved, the applicatio server continue service for the browser. The Web server which be loaded Dispatcher responds the client's request. This flow is as Figure 1. The hit of each step is following.

Step 1: Browser1 send a visit request to the Web server.

Step 2: Dispatcher in the Web server does not search cookie in the browser1. Here the Browser1 access the Web server at first time. So, the Web server balance the load on each application server in the server cluster according the configure documents and send the request to the application server which's load is easier, now we suppose to it is App_1.

Step 3: App_1 responds to the request and creates a Session for Borwser1. App_1 encodes own IP and Port address to B64 code and writes the code and Session ID to the cookie. Then App_1 send the message to the Web server

Step 4: The Web server refreshes Browser1.

Step 5: Browser1 requests the other page before the Session in App_1 server is not overtime.

Step 6: Web Dispatcher read the B64 code in the cookie and matches it with the B64 code in the configure document in the application servers. So the Web server can confirm which application server should respond to the browser. Now we suppose which is App_1. The Web server transmits the Browser1 request to App_1.

Step 7: While the Browser1's request for App_1 is no over, the Browser2 sends a request to the Web server.

Step 8: Dispatcher confirms that there is not the cookie, and will turn the request to another application server, now it is App_2.

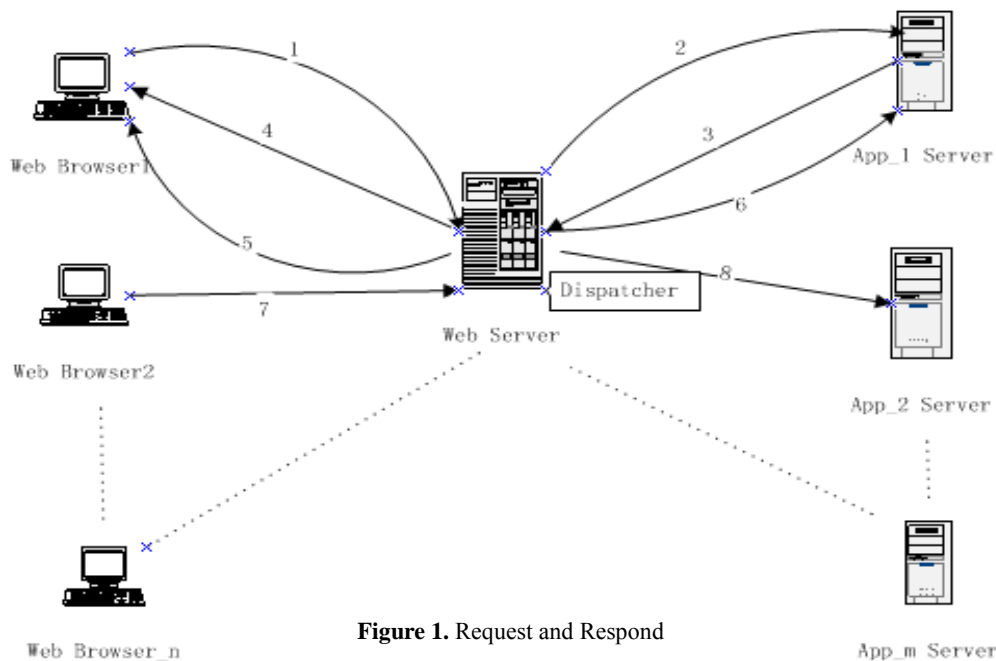


Figure 1. Request and Respond

Supervise Dispatcher status	Web communicates and process client request
	Read config document and interpret XML
	Load balancing
	Dispatcher
	Analysis of application server status
	Thunderbolt process
	Socket message

Figure 2. Dispatcher Structure

According to the place that Dispatcher is kept, we can confirm the basic functions are load balancing and dispatcher. The strategy for load balancing can be set up by the system administrator. Here we set up the strategies are the simple circulation and the recurrence with scale. The dispatcher ensures that the Session keep on response with an application server according to the B64 code in the cookie. As the function of Dispatcher, load balancing and dispatcher need support from the configure document.

So, Dispatcher should be able to read and interpret the configure document in XML. Before balancing load and

dispatcher, it is must to read the status of the application server and scale value. So, it is important to analyze the status of application server. In addition, Dispatcher should be able to handle exited. As the Web server communication with the browser and client request process are the key, the Socket communication part is necessary. For implementing all above, Dispatcher need a supervise part, this is the status supervise of the Dispatcher

3. PARTS CODE OF DISPATCHER

3.1 Read and Interpret the Configure Document

The configure document is in the XML. It is divided into three parts: <Options>, <Application> and <Status>.

```
<Dispatcher Configure>
.....
<Options retry_time="30" />
.....
<Application prefix="/demoApp">
<AppServer host="host1" port="7000" weight="1"
auth="myPass" />
<AppServer host="host2" port="7001" weight="2"
auth="myPass" />
<AppServer host= ..... />
<Restrict client="10.20.20.0/24" />
</Application>
.....
<Application prefix="/simpleApp">
<AppServer host="host3" port="7009" />
<AppServer ..... />
</Application>
.....
<Status prefix="/status">
<Restrict server="127.0.0.1" />
<Restrict client="127.0.0.1" />
</Status>
.....
</Dispatcher Configure>
```

According to the prefixes, Dispatcher distinguishes each application (the number for the all application is not limited.). Set up the strategies for the load balancing are the simple circulation and the recurrence with scale. Use the <AppServer> label to describe each server. <Status> is used to supervise and control the real time status of Dispatcher. <Restrict> label describe the special restriction to client by the Web server. The configure document will be interpreted by the DOM tree way. For example:

```
edir_conf_t *configure=NULL; // definite tree structure
for (node=pnode->children; node; node = node->next)
{
    if (!edir_strcasecmp((char *) node->name, "Application"))
        numApps++;
    if (!edir_strcasecmp((char *) node->name, "Status"))
        numStats++;
}
/*count to the < Application> son node and <Status> son
node*/
Config-> apps= edir_conf_apps_create( numApps);
// define the structure unit by number
Config-> stats= edir_conf_stats_create( numStats);
// define the structure unit by number
for (node=pnode->children; node; node = node->next)
{
    if (!edir_strcasecmp((char *) node->name, "Options"))
        edir_conf_xml_walk_options(node, config, exc);
    // get value of the option son node
    if (!edir_strcasecmp((char *) node->name, "Application"))
        config->apps[numApps++] =
            edir_conf_xml_walk_application(node, exc);
    //get value of the e Application son node
    if (!edir_strcasecmp((char *) node->name, "Status"))
```

```
config->stats[numStats++ =
            edir_conf_xml_walk_status(node, exc);
    //get value of the status son node
}
```

3.2 Load Balancing

```
<Application prefix="/Test">
<AppServer host=" host1" port="9001"/> --- " Server1"
<AppServer host=" host1" port="9002"/> --- " Server2"
<AppServer host=" host2" port="9001"/> --- " Server3"
<AppServer host=" host2" port="9002"/> --- " Server4"
</Application>
```

To carry the Dispatcher on the Web server, the client request is possibly reasonable to load every application server. Here, host1 and host2 is the name of two servers. For the same machine, it can be circulated different or same services on the different port. Here we announce of are all Test services. Must notice: "the server" is an application that is announced at a port.

Implementing of load balancing round with scale is easy in fact: Compute the score of each server, and make the server that which has the max score to respond to the client request. But must notice a few problems: If the link count to 0, then let this server has max score. It is that the round with scale degenerate the simple round. Pseudo code of load balancing is following.

```
Let maxWeight=max_weight which application servers have;
Let MAXINT= 0xFFFFFFFF; //unsigned long int 32 bit
Let factor = MAXINT / maxWeight;
Let maxScore = -1;
Let maxServer = nil;
For server in {all server which circulate same application of }
    // search for in all servers
    Let numConn={ the link number of the current server};
    // obtain the link number of the current server
    Let weight={ scale value of the current server};
    if numConn > 0 then // if server has link
        score = (weight * factor) / numConn;
        // compute score according to the formula
    else
        score = MAXINT;
    end if;
    if score > maxScore then
        maxScore = score;
        maxServer = server;
        // make the current server to the candidate server
    end if
end For
if (maxServer not nil) return maxServer
// the Server is selected, and return the Server
```

Finally, exit handle will return the extraordinary value and name of extraordinary text. Status supervisor calls basic function to realize the supervision e and control to the module.

4. REFERENCES

- [1] G. Holden, N. Wells, Matthew Keller, Apache Server, Beijing: Machine Industry Publishing Co., 2001
- [2] Mohammed J. Kabir, Practice to Apache Server, Beijing: Electron Industry Publishing Co., 2002

- [3] H. Bryhni, E.Kloving, O.Kuve, “ A comparison of Load Balancing Technologies for Scalable Web Servers”[J] IEEE Network 2000.7/8 pp.56~63
- [4] Wang Yunlan, Li Zengzhii, Xun Jun, Ban Shimin, “ The Algorithms of the DNS- Based Load Balancing System”,[J] Computer Engineering and Application, November 2002, pp.11~13.
- [5] Liu Qing- rui, “Research on Loading Balancing of the Middleware”,[J] Mini- Micro System, Vol.23 No.3 Mar 2002, pp.374~376.
- [6]Chen Zhi- gang, Li Deng, Zeng Zhi- wen, “ The Dynamic Load Balancing Implementation Model in The Distributed System”[J] CENT.SOUTH UNIV. Technology, Vol.32 No.6 December 2001,pp.625~639



Tang Wei is an Associate Professor, in Math and Computer Department, Jiangnan University. His research interests are in information system analysis and design, information security. He has published two textbooks, more than ten Journal papers.

Finish Time Maximization Method: an Anti-Sequence Algorithm to Scheduling Task Graphs for Multiprocessors

Jun Sun, Wenbo Xu, Bin Feng

School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu Province 214036, China PR

Email: sunjun21c@163.com

ABSTRACT

The problem of task scheduling on multiprocessors is NP-complete for most cases. Because of the intractability of the problem, heuristic ideas are used in most of the existed algorithms. In this paper, we propose an anti-sequence scheduling algorithm, FTM algorithm, of which the mapping strategy is based on an anti-topology order. After the description of the algorithm, we present an illustrating example and the experiment results for a group of task graphs. It is shown that the solution quality acquired by the proposed algorithm outperforms some other sequence scheduling algorithms.

Keyword: task graph, scheduling, multiprocessor, sequence scheduling, anti-sequence scheduling

1. INTRODUCTION

Because of its NP completeness, the tasking scheduling problem on multiprocessors is often tackled by appealing to heuristic techniques, among which is list scheduling. List scheduling is often used for a bounded number of processors (e. g., MCP [5], DPS [8]), and there are usually two steps in a list-scheduling algorithm. That is, (1) acquire a scheduling list by some priority strategy; (2) remove every node and schedule them to the target processors by the order of the scheduling list.

As of step (1) of scheduling, almost all of the existing algorithms usually generate a topology order constrained by the DAG of the task, and therefore, the second step, is executed to remove and schedule the nodes by the order of from the first node to the last one in the topology order. We called these strategies sequence-scheduling algorithms (SSA). In this paper, we present an anti-sequence algorithm, called Finish Time Maximization (FTM) Step, in which the input of the second step is an anti-topology order instead of a topology order. The experiments show that our proposed algorithm has much advantage over the SSAs in many cases, particularly when the DAG of a parallel task has more than one exit node.

The remaining paper is organized as follows: Section 2 defines the scheduling problem and introduces some definitions used in the paper. Section 3 describes our proposed algorithm. Section 4 includes a scheduling example illustrating the operation of the algorithm. Next, we present the experiment results in Section 5, and conclude the paper with some remarks in Section 6.

2. PROBLEM STATEMENT

In static scheduling problem on multiprocessors, a parallel program is often modeled by a directed acyclic graph (DAG)

$G=(V, E)$, where V is a set of v nodes, representing the tasks, and E is a set of directed edges, representing the communication message. Edges in a DAG are directed and, thus, determine the precedence constraints among the tasks. The cost of node n_i , denoted as $w(n_i)$, represents the computation cost of the task. The weight of an edge (n_i, n_j) , denoted by $c(n_i, n_j)$, represents the communication cost of the message. The source node of an edge is called a parent node, while the destination node is called a child node. A node with no parent node is called an entry node and a node with no child is called an exit node. The precedence constraints of a DAG dictate that a node can only start execution after it has gathered all of the messages from its parent nodes. An example of DAG, shown in Fig. 1, will be used as an example in the subsequent discussion.

The communication to computation ratio (CCR) of a task graph is a measure of the task graph granularity and can be defined in various ways. We adopted the definition used in [2] which defines CCR as the ratio between the average communication and computation costs in the task graph. The *bottom level (b-level)* of node n_i , denoted as $bl(n_i)$, is defined as the length of the longest path from that node to an exit task, while the *top level (t-level)* of node n_i , denoted as $tl(n_i)$, is the length of the longest path from an entry node to that node (excluding the cost of that node). These two quantities can be computed by the following recurrent formulations.

$$\begin{cases} tl(n_1)=0 \\ tl(n_i)=\max_{n_j} \{tl(n_j)+w(n_j)+c(n_j, n_i)\}, n_j \in pred(n_i), i=1,2,\Lambda v \end{cases} \quad (1)$$

where $pred(n_j)$ is the set of n_j 's parent nodes and n_1 is the entry node of the DAG.

$$\begin{cases} bl(n_v)=w(n_v) \\ bl(n_i)=\max \{bl(n_k)+w(n_i)+c(n_i, n_k)\}, n_k \in succ(n_i), i=1,2,\Lambda v \end{cases} \quad (2)$$

where $succ(n_i)$ is the set of n_i 's child nodes and n_v is the exit node of the DAG.

When using the formulations to obtain t-level and b-level of a node, we must exploit topological sort algorithm and anti-topological algorithm. Generally, the objective of scheduling is to minimize the schedule length, which defined as

$$SL = \max_i \{FT(n_i)\} \quad (3)$$

across all nodes, by proper allocation of the nodes to the processors and arrangement of execution sequencing of the nodes without violating the precedence constraints. Table 1

summarizes the definitions of the notations used in the paper.

The target system is also assumed to be a fully connected distributed memory multiprocessor system with no regard to link contention and scheduling of message. An example of processor graph is shown in Fig. 2.

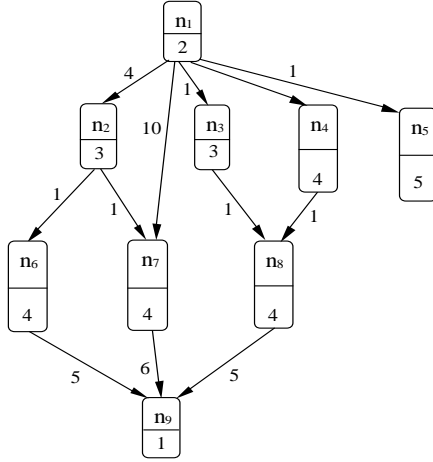


Fig. 1. An Example of Task Graph

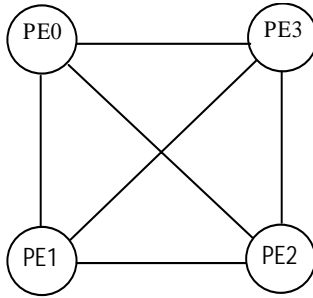


Fig. 2. A 4-processor fully connected target system

Table 1. Definitions of Notations

NOTATION	DEFINITION
v	The total number of nodes in the task graph
e	The total number of edges in the task graph
n_i	A node in the parallel program task graph
$w(n_i)$	The computation cost of node n_i
$c(n_i, n_j)$	The communication cost of the directed edge from node n_i to n_j
p	Number of processors
CP	A critical path of the task graph
CPN	Critical path node
$DAT(n_i)$	The possible data available time of node n_i
$ST(n_i)$	The start time of node n_i
$FT(n_i)$	The finish time of node n_i
CCR	Communication-to-computation Ratio
IBN	In-branch node
OBN	Out-branch node

3. THE PROPOSED ALGORITHM

As mentioned in Section one, our proposed anti-sequence

scheduling algorithm uses an anti-topology order as a scheduling list. It is different from the sequence scheduling algorithm in that the last node in the topology order is scheduled first and the finish time of node on a processor is evaluated. To compute $FT(n_i, Q)$, the finish time of node n_i on processor Q , the computation of $DST(n_i, Q)$, the possible data sending time of the node, is also essential. The following is rule of the computation of $FT(n_i, Q)$ and $DST(n_i, Q)$.

Rule 1: Computation of $FT(n_i, Q)$:

If m nodes $\{n_{Q_1}, n_{Q_2}, \dots, n_{Q_m}\}$ have been scheduled on processor Q ($m \geq 0$), the finish time of n_i on processor Q is

$$FT(n_i, Q) = \max\{ST(n_{Q_m}, Q), DST(n_i, Q)\} \quad (4)$$

Rule 2: Computation of $FT(n_i, Q)$:

Precondition: m nodes $\{n_{Q_1}, n_{Q_2}, \dots, n_{Q_m}\}$ have been scheduled on processor Q ($m \geq 0$).

1. Check if there exists some k such that:

$$FT(n_{Q_{k+1}}, Q) - \max\{ST(n_{Q_k}, Q), DST(n_i, Q)\} \geq w(n_i) \quad (5)$$

where $k = 0, K, m, ST(n_{Q_{m+1}}, Q) = \infty$, and $FT(n_{Q_0}, Q) = 0$.

2. If such k exists, compute $FT(n_i, Q) = \max\{ST(n_{Q_l}, Q), DST(n_i, Q)\}$ with l being the smallest k satisfying the above inequality, and return this value as the start time of n_i on processor Q ; otherwise, return ∞ . After computation of $FT(n_i, Q)$, we can work out $ST(n_i, Q)$ and $DST(n_i, Q)$ by

$$ST(n_i, Q) = FT(n_i, Q) - w(n_i) \quad (6)$$

$$DST(n_i, Q) = \max_{n_j} \{FT(n_j) - w(n_j) + c(n_j, n_i)\}, n_j \in \text{succ}(n_i) \quad (7)$$

Using the above rule, we can get the schedule length of the problem by

$$SL = \max\{FT(n_i)\} - \min\{ST(n_i)\} \quad (8)$$

where $\max\{FT(n_i)\}$ and $\min\{ST(n_i)\}$ denote the maximum of the finish time and minimum of the start time cross all the nodes, respectively. Our proposed anti-sequence scheduling algorithm is described by the following.

Finish Time Maximization Algorithm (FTM Algorithm)

- (1) Input a topology order as a scheduling list $sl[v]$;
- (2) **for** $I=v-1$ to 0 **do**
- (3) $Max_FT=0$;
- (4) **for** $j=0$ to $p-1$ **do**
- (5) compute $FT(n_x, P_j)$ according to Rule 1 or Rule 2;
- (6) **if** $FT(n_x, P_j) > Min_FT$ **then** $Max_FT = FT(n_x, P_j)$

- (7) let P_j be the target processor;
- (8) **endif**
- (9) **endfor**
- (10) schedule the n_x onto the target processor;
- (11) **endfor**
- (12) adjust the schedule to make its start time be zero;
- (13) output the outcome;

Analyzing the FTM algorithm, we can find out that the time complexity of the algorithm is $O(vp)$ or $O(vvp)$ if we use Rule 1 or Rule 2 to compute $FT(n_i, Q)$ respectively. The latter has better performance, whereas the former has less running time.

4. AN ILLUSTRATING EXAMPLE

In this section, we use Fig. 1 for an illustrating example of FTM algorithm. The outcome result of a sequence-scheduling algorithm, start time minimization algorithm (STM) is shown in Fig. 3. And the result of FTM algorithm is shown in figure 4. This example shows that although the scheduling length is the same in both algorithms, the schedule resulting from FTM employs less processors than that from STM algorithm.

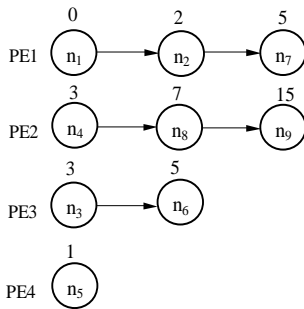


Fig. 3. The schedule generated by STM algorithm

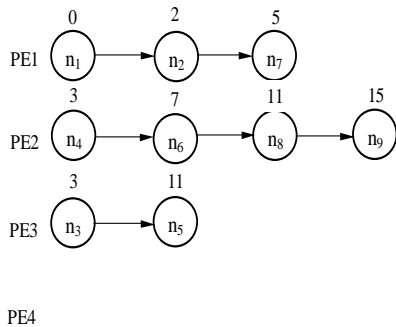


Fig. 4. The schedule resulted from FTM algorithm

5. EXPERIMENT RESULTS

Because of the NP-completeness of scheduling problem, heuristic ideas used in FTM algorithm cannot always lead to an optimal solution. Thus, it is necessary to compare the average performance of different algorithms by using randomly generated graphs. Figure 5 is the comparison of the schedule length for a group of task graphs using FTM and STM. Figure 6 is the comparison of the number of processors employed in

the two cases.

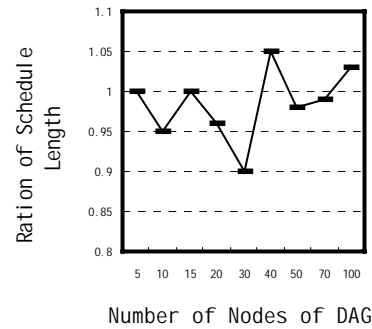


Fig. 5. Ration of Schedule Length of FTM and STM

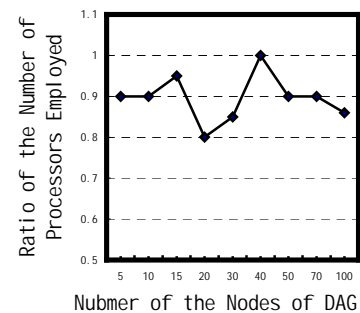


Fig. 6. Ration of the Number of Processors employed

6. CONCLUSIONS

In this paper, we have presented a new list-scheduling algorithm, which is based on anti-topology order and has good performance compared with other algorithms. The proposed algorithm may generate a better schedule than other sequence scheduling algorithm because the schedule employs less processors, particularly when a DAG has more than one exit node. Fortunately, in practice, the DAG of a parallel program usually has at least two exit nodes, so the FTM algorithm could work better than other algorithm.

7. REFERENCES

- [1] Andrei Radulescu and Arjan J.C. van Gemund, "Low-Cost Task Scheduling for Distributed-Memory Machines", IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No.6, June 2002, pp648-658.
- [2] Yu-Kwong Kwok and Ishfaq Ahmad, "Benchmarking and Comparison of the Task Graph Scheduling Algorithms", Journal of Parallel and Distributed Computing 59, 381-422 (1999)
- [3] Ishfaq Ahmad and Yu-Kwong Kwok, "On Parallelizing the Multiprocessor Scheduling Problem", IEEE Transactions on Parallel and Distributed System, Vol. 10, No. 4, April 1999, pp414-432.
- [4] M. R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., 1979.
- [5] Min-you Wu and Daniel D. Gajski, "Hypertool: A Programming Aid for Message-Passing Systems", IEEE

- Transactions on Parallel and Distributed Systems, Vol. 1, No. 3, July 1996.
- [6] Tao Yang and Apostolos Gerasoulis, "DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors", IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 9, September 1994, pp-951-967.
 - [7] Yu-Kwong Kwok and Ishfaq Ahmad, "Dynamic Critical-Path Scheduling: An Effective Technique for Allocating Task Graphs to Multiprocessors", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No.5, May 1996.
 - [8] G. -L. Park, B. Shirazi, J. Marquis, and H. Choo, "Decisive Path Scheduling: A New List Scheduling Method", Proc. Int'l Conf. Parallel processing (ICPP), Aug. 1997, pp472-480.
 - [9] M. A. Palis, J.-C. Liou, and D.S.L. Wei, "Task Clustering and Scheduling for Distributed Memory Parallel Architectures", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No.1, January 1996, pp46-55.
 - [10] S.J. Kim and J.C. Browne, "A General Approach to Mapping of Parallel Computation Upon Multiprocessor Architectures", Proc. Int'l Conf. Parallel Processing (ICPP), Vol. 3, Aug. 1988, pp1-8.

Implementing and Invoking a Remote Object Calling Native Methods via RMI-IIOP and JNI

Minglong QI, Qingping GUO, Luo ZHONG
School of Computer Science, Wuhan University of Technology
Ma Fang San Campus, 430070 Wuhan, China
Email: minglongqi@sina.com; Tel.: +86-(0)27-87292452

ABSTRACT

In this article, we discuss how to combine one of the distributed computing technologies created by Sun Microsystems, called RMI over IIOP, and its other famous technology, Java Native Interface JNI. Sometimes in a distributed computing application written entirely in Java, client side application need to invoke native methods specific to a platform and encoded in a another programming language (for example in C or C++) .To do this, we must make a distributed computing solution that can integrate native applications in Java. We think that Sun's RMI over IIOP and JNI is a good choice. The first reason is that, RMI over IIOP takes advantage of both RMI (easy to use and encoding uniquely in Java), and IIOP (interoperability with another CORBA ORB products). In addition to this, we can switch transport protocols from JRMP (Java Remote Method Protocol) to IIOP (Internet Inter-ORB). The second reason is that, JNI cannot only integrate native applications in Java, but can also embed JVM implementation in a native application. To demonstrate the combination between RMI-IIOP and JNI, we have developed a typical example: on the server side, implementation of a remote interface method create an instance of a Java class and called its native method, whose implementation use JNI and Borland InterBase C API to extract a database table. On the client side it invoked this remote object and displays the database table.

Keywords: RMI-IIOP, JNI, InterBase API, CORBA, ORB.

1. INTRODUCTION

RMI-IIOP is a distributed computing technology uniquely using Java language for encoding, in which we do not need to use IDL (Interface Definition Language) to establish interfaces. This does not means IDL interface is completely excluded from RMI-IIOP. In fact we can generate the same interface IDL version from the Java-encoded interface by using -idl option of rmic command, where it is possible to operate with another ORB products. Many tutorials and articles have come out to treat this subject, but applications or techniques discussed there have been always focused in Java language and in Java remote objects. In this distributed computing architecture, people are seduced by the elegance of Java's "written once, run anywhere" feature and forget to go out of the Java world. We asked ourselves the question: Can a RMI-IIOP client invoke a remote method that in its turn calls a native method situated in server side? The answer resides in the combination between RMI-IIOP and JNI. Legitimately, it is JNI that provides the possibility of integration of platform-specific native codes or libraries in Java. JNI is a integral part of JVM implementation, and by doing this there is a loss of the Java cross-platform feature but the benefit gained the flexibility in the software development. There are also many tutorials and articles that treat JNI subject, but not many link it to distributed computing.

We have tried to make a bridge between those two technologies through a concrete example: a RMI-IIOP client invokes a remote object that calls a native method implemented by JNI and Borland InterBase C API for extracting and displaying a database table. We call this "remote native method invocation".

2. BRIEF DESCRIPTION OF RMI-IIOP AND JNI

In books or tutorials about this subject, RMI-IIOP is descriptively defined below: Java Remote Method Invocation ("Java RMI") technology run over Internet Inter-Orb Protocol ("RMI-IIOP") delivers Common Object Request Broker Architecture (CORBA) distributed computing capabilities to the Java 2 platform. Java RMI over IIOP was developed by Sun and IBM.

Java RMI over IIOP combines the best features of Java RMI technology with the best features of CORBA technology. Like Java RMI, RMI over IIOP speeds distributed application development by allowing developers to work completely in the Java programming language. When using RMI over IIOP to produce Java technology-based distributed applications, there is no separate Interface Definition Language (IDL) or mapping to learn. Like RMI, RMI over IIOP provides flexibility by allowing developers to pass any serializable Java object (Objects By Value) between application components. Like CORBA, RMI over IIOP is based on open standards defined with the participation of hundreds of vendors and users in the Object Management Group. Like CORBA, RMI over IIOP uses IIOP as its communication protocol. IIOP eases legacy application and platform integration by allowing application components written in C++, Smalltalk, and other CORBA supported languages to communicate with components running on the Java platform.

RMI over IIOP is based on two specifications of the Object Management Group: Java Language Mapping to OMG IDL Specification and CORBA/IIOP 2.3.1 Specification, formal/99-10-07. With RMI over IIOP, developers can write remote interfaces in the Java programming language and implement them just using Java technology and the Java RMI APIs. These interfaces can be implemented in any other language that is supported by an OMG mapping and a vendor supplied ORB for that language. Similarly, clients can be written in other languages using IDL derived from the remote Java technology-based interfaces. Using RMI over IIOP, objects can be passed both by reference and by value over IIOP.

The Java Native Interface (JNI) is a powerful feature of the Java platform. Applications that use the JNI can incorporate native code written in programming languages such as C and C++, as well as code written in the Java programming language. The JNI allows programmers to take advantage of the power of

the Java platform, without having to abandon their investments in legacy code. Because the JNI is a part of the Java platform, programmers can address interoperability issues once, and expect their solution to work with all implementations of the Java platform. As a two-way interface, the JNI can support two types of native code: native libraries and native applications. The JNI supports an invocation interface that allows you to embed a Java virtual machine implementation into native applications. Native applications can link with a native library

that implements the Java virtual machine, and then use the invocation interface to execute software components written in the Java programming language. For example, a web browser written in C can execute downloaded applets in an embedded Java virtual machine implementation.

3. A EXAMPLE

Our problem is as follows: How to create a RMI over IIOP distributed computing application that allows the extraction of a database table from the server side and display the same database table in the client side by using RMI API, JNI and Borland Interbase C API? We have developed this software under Windows 2000. The tools used in the development are Borland JBuilder 6.0, Borland InterBase 5.6 and Borland C++ 5.0 IDE. The reason we choose Borland's products was that InterBase C API is required to implement the native method. For other operating systems such as Red Hat Linux 8.0, the development of the software should follow in a similar way.

Create JNI part of the application

A simple JNI application is created through the following stages: creating a Java class where some native methods are declared and where a static initializer must be declared for loading the library containing implementation of the native methods; Compiling the Java class using javac command; Generating the JNI C-style head file by using javah command with -jni option; Implementing the native methods using JNI API and another C API (in our case, it is InterBase C API); Linking C object files to generate a dynamic linking library (DLL) by using Visual C++ linker or Borland C++ linker; Running the application using java command with the option -Djava.library.path = path of the dynamic link library of the application. Creating a project in the directory C:\rmi\RemoteNativeMethod with JBuilder 6.0, and creating a java class that declares a native method and has a static initializer to load the library, the code of this Java class is below (HaveNativeMethods.java) :

Table 1. Code of the Java class that declares a native method

```
package remotenativemethod;
public class HaveNativeMethods {
/*Native method to be implemented for extracting a database
table from InterBase */
public native String[][] getDBTable();
/*Static initializer to load the Dynamic Link Library named
NativeDBLib */
static {
    System.loadLibrary("NativeDBLib");
}
}
```

The native method public native String[][] getDBTable(); will be implemented by using JNI functions and Borland InterBase C API. The return value String[][] represents a database table. Rebuilding it with option -jni by JBuilder results in a JNI head file, whose code is below (remotenativemethod_HaveNativeMethods.h):

Table 2. Code of the generated JNI C-style head file

```
/* DO NOT EDIT THIS FILE - it is machine generated */
#include <jni.h>
/* Header for class remotenativemethod_HaveNativeMethods
*/

#ifndef _Included_remotenativemethod_HaveNativeMethods
#define _Included_remotenativemethod_HaveNativeMethods
#ifdef __cplusplus
extern "C" {
#endif
/*
 * Class:         remotenativemethod_HaveNativeMethods
 * Method:        getDBTable
 * Signature:     ()[[Ljava/lang/String;
 */
JNIEXPORT jobjectArray JNICALL
Java_remotenativemethod_HaveNativeMethods_getDBTable
(JNIEnv *, jobject);

#ifdef __cplusplus
}
#endif
#endif
```

Now come the most important and the most difficult stage in the JNI development: implementing the native method. We used some InterBase C API and JNI functions for programming extraction of a database table. This is a tutorial database table in InterBase, It is phone list, where we want just to extract three of the fields: last_name, first_name and phone_ext for people of Monterey city. You must include the generated JNI C-style head file "remotenativemethod_HaveNativeMethods.h" and InterBase C API head file <ibase.h>. The data structure XSQlda in InterBase C API is data type for holding the result of a SQL query on a table, similar to java.sql.ResultSet in JDBC API. A key problem is how to copy the data set obtained from InterBase C API data structure to JNI data structure. In the Java programming language, a two-dimension String type array can hold the data set of a database table, such as an array of String[], that corresponds to a two-dimension object array in JNI . The next code snippet demonstrates how to create a two-dimension jstring type object array:

```
... ..
jobjectArray resultSet, oneRecord;
jclass stringCls, objCls;
jint nbFields, nbRecords;
stringCls = (*env)->FindClass(env, "java/lang/String");
if (stringCls == NULL) { ... }
oneRecord = (*env)->NewObjectArray(env, nbFields ,
stringCls, NULL);
if (oneRecord == NULL) { ... }
/*Get the class reference for object oneRecord*/
objCls = (*env)->GetObjectClass(env, oneRecord);
if (jstringCls == NULL) { ... }
/*Finally construct two-dimension object array*/
```



```

resultSet=
(*env)->NewObjectArray(env,nbRecords,objCls , NULL);
... ..

```

For people who are not familiar to InterBase C API or JNI programming, please refer to InterBase API Programmer's Guide or JNI Programmer's Guide. The complete implementation code is in a C source file named `remoteNativeMethodImpl.c`, shown below:

Table 4. Code of implemetation of the native method

```

#include "remotenativemethod_HaveNativeMethods.h"
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <ibase.h>
#define LASTLEN 20
#define FIRSTLEN 15
#define EXTLEN 4
JNIEXPORT jobjectArray JNICALL
Java_remotenativemethod_HaveNativeMethods_getDBTable
(JNIEnv *env, jobject obj)
{
    char last_name[LASTLEN + 2];
    char first_name[FIRSTLEN + 2];
    char phone_ext[EXTLEN + 2];
    short flag0 = 0, flag1 = 0;
    short flag2 = 0;
    /* statement handle */
    isc_stmt_handle stmt = NULL;
    /* database handle */
    isc_db_handle DB = NULL;
    /* transaction handle */
    isc_tr_handle trans = NULL;
    /* status vector */
    long status[20];
    /*hold the result data set of the query */
    XSQLDA ISC_FAR * sqllda;
    long fetch_stat;
    /*database name employee.gdb by default */

    char empdb[128];
    char *sel_str =
        "SELECT last_name, first_name, phone_ext FROM
phone_list \
        WHERE location = 'Monterey' ORDER BY
last_name, first_name;";

    /* declaration for JNI variables*/
    jobjectArray oneRecord,tempRecord,resultSet;
    jclass stringCls,objCls;
    int i,j;
    jstring str1,str2,str3;
    strcpy(empdb, "employee.gdb");

    /*connect to database*/
    if (isc_attach_database(status, 0, empdb, &DB, 0, NULL))
isc_print_status(status);
    /*create a transaction*/
    if (isc_start_transaction(status, &trans, 1, &DB, 0,
NULL)) isc_print_status(status);
    /*Allocate an output SQLDA.*/
    sqllda = (XSQLDA ISC_FAR *)
malloc(XSQLDA_LENGTH(3));

```

```

sqllda->sqln = 3;
sqllda->sqld = 3;
sqllda->version = 1;
/* Allocate a statement.*/
if (isc_dsql_allocate_statement(status, &DB, &stmt))
isc_print_status(status);
/* Prepare the statement.*/
if (isc_dsql_prepare(status, &trans, &stmt, 0, sel_str, 1,
sqllda)) isc_print_status(status);

sqllda->sqlvar[0].sqldata = (char ISC_FAR *)&last_name;
sqllda->sqlvar[0].sqltype = SQL_TEXT + 1;
sqllda->sqlvar[0].sqlind = &flag0;

sqllda->sqlvar[1].sqldata = (char ISC_FAR *)&first_name;
sqllda->sqlvar[1].sqltype = SQL_TEXT + 1;
sqllda->sqlvar[1].sqlind = &flag1;

sqllda->sqlvar[2].sqldata = (char ISC_FAR *) phone_ext;
sqllda->sqlvar[2].sqltype = SQL_TEXT + 1;
sqllda->sqlvar[2].sqlind = &flag2;

/* Execute the statement.*/
if (isc_dsql_execute(status, &trans, &stmt, 1, NULL))
isc_print_status(status);

/*Create JNI data objects for holding the result of query*/
stringCls = (*env)->FindClass(env, "java/lang/String");
if (stringCls == NULL)return NULL;

oneRecord = (*env)->NewObjectArray(env, 3, stringCls,
NULL);
if (oneRecord == NULL) return NULL;

objCls = (*env)->GetObjectClass(env,oneRecord);
if (jstrCls == NULL)return NULL;

/*Suppose there are 200 records*/
resultSet = (*env)->NewObjectArray(env, 200,objCls ,
NULL);
if (resultSet == NULL) return NULL;

/*
 * Fetch and print the records.
 * Status is 100 after the last row is fetched.
 */
i=0;
while ((fetch_stat = isc_dsql_fetch(status, &stmt, 1, sqllda))
== 0)
{
    last_name[sqllda->sqlvar[0].sqln] = '\0';
    first_name[sqllda->sqlvar[1].sqln] = '\0';
    phone_ext[sqllda->sqlvar[2].sqln] = '\0';

    tempRecord = (*env)->NewObjectArray(env, 3,
stringCls, NULL);
    if (tempRecord == NULL) return NULL;

    str1=(*env)->NewStringUTF(env,last_name);
    (*env)->SetObjectArrayElement(env,tempRecord,0,str1);
    str2=(*env)->NewStringUTF(env,first_name);
    (*env)->SetObjectArrayElement(env,tempRecord,1,str2);

```

```

        str3=(*env)->NewStringUTF(env,phone_ext);
(*env)->SetObjectArrayElement(env,tempRecord,2,str3);
        if (fetch_stat == 0)
        {
(*env)->SetObjectArrayElement(env,resultSet,i,tempRecord);
        (*env)->DeleteLocalRef(env,tempRecord);
        i++;
        }
        if (fetch_stat == 100L)

(*env)->SetObjectArrayElement(env,resultSet,i,NULL);
    }

    if (fetch_stat != 100L)    isc_print_status(status);

    /* Free statement handle. */
    if (isc_dsql_free_statement(status, &stmt, DSQL_close))
isc_print_status(status);

    /*Commit the transaction*/
    if (isc_commit_transaction(status, &trans))
isc_print_status(status);

    /*Disconnect from database*/
    if (isc_detach_database(status, &DB))
isc_print_status(status);

    /*Free InterBase API result data set*/
    free( sqlda);
    return resultSet;
}

```

The next stage is to build a makefile for Borland C++ compiler and linker, and this is not so easy. In the INCLUDE macro, you must put JNI API path. It is situated in the JDK's include directory. It is also necessary to put InterBase C API path in the INCLUDE macro. This is situated in the InterBase include directory. You can get bcc32 compiler options infos by typing bcc32 under DOS. The bcc32's options used in this makefile are : -c Compile only; -v Source level debugging; -a4 Align on 4 bytes; -tWM Makes the target multi-threaded; -DWIN32 Defines the string "WIN32"; -tWC Makes the target a console .EXE with all functions exportable; -tWCDE Makes the target a console .DLL with all functions exportable; -w Turn warning control off. For Borland linker tlink32 option infos, you can obtain them by typing tlink32 command under DOS. tlink32's options used in this makefile are: /c Case sensitive link; /x No map; /ap Windowing compatible; and /Tpe Output file being .EXE type; and /Tpd Output file being .DLL type. And finally you must link the application with InterBase library gds32.lib. The code of the makefile is shown below:

Table 5. Code of makefile for Borland compiler

```

IBASE=      C:\InterBase
JBDIR=      C:\JBuilder6\jdk1.3.1
BCDIR=      C:\BC5
COMMON_FLAGS=  -c -v -w- -a4 -tWM -DWIN32
$(INCLUDE)
CFLAGS=      $(COMMON_FLAGS) -tWC
LIB_CFLAGS=   $(COMMON_FLAGS) -tWCDE
INCLUDE=      -I$(IBASE)\include -I$(BCDIR)\include
-I$(JBDIR)\include -I$(JBDIR)\include\win32
LFLAGS=      /c /x /ap /Tpe

```

```

LIBS=        $(IBASE)\lib\gds32.lib
CC=          $(BCDIR)\bin\bcc32
LINK=        $(BCDIR)\bin\tlink32
IMPLIB=      $(BCDIR)\bin\implib

```

all: NativeDBLib.dll

```

remoteNativeMethodImpl.obj:    remoteNativeMethodImpl.c
remotenativemethod_HaveNativeMethods.h

```

\$(CC) \$(LIB_CFLAGS) remoteNativeMethodImpl.c

```

NativeDBLib.dll: remoteNativeMethodImpl.obj
@echo $(BCDIR)\lib\c0d32.obj+ > link.arg
@echo    remoteNativeMethodImpl.obj >>

```

link.arg

```

@echo $@ >> link.arg
@echo /x /Tpd >> link.arg
@echo $(LIBS)+ >> link.arg
@echo $(BCDIR)\lib\import32.lib+ >> link.arg
@echo $(BCDIR)\lib\cw32mt.lib >> link.arg
$(LINK) @link.arg

```

Attention to the options of make command must use -l option to enable use of long command lines. For constructing our dynamic linking library named "NativeDBLib.dll", you just need to type the next command "make -l -f makefile NativeDBLib.dll" under DOS.

Create RMI-IIOP part of the application and combine RMI-IIOP and JNI

The construction of a distributed application of RMI over IIOP type is as follows: define a remote interface that extends the standard interface java.rmi.Remote, where every declared method must throw the java.rmi.RemoteException; create a Java class that extends javax.rmi.PortableRemoteObject and implements your remote interface, and the constructor of this class must throw the java.rmi.RemoteException; create server application where an instance of the remote interface implementation class must be generated and bound to a naming context object by name; and finally create client application where a remote object will be looked up from a naming context object by name, and the remote method will be invoked via this remote object. In our remote interface (shown in table 6), we define a remote method called "public String[][] invokeRemoteNativeMethod()" throws java.rmi.RemoteException; ", that will call a native method in its implementation for extracting the database table phone list.

Table 6. Code of the remote interface RemoteNativeMethodInterface.java

```

package remotenativemethod;
public interface RemoteNativeMethodInterface extends
java.rmi.Remote {
public String[][] invokeRemoteNativeMethod() throws
java.rmi.RemoteException;
}

```

The code of the Java class that implements the remote interface is given in table 7. Notice that implementation of the remote method invokeRemoteNativeMethod() creates an object of the class HaveNativeMethods, Via this object, the native method getDBTable() is called to return the data set of the database table phone_list. The code for server and client is shown

repectively in table 8 and in table 9.

Table 7. Code of the Java class that implements the remote interface

```
package remotenativemethod;
import javax.rmi.PortableRemoteObject;
import java.rmi.RemoteException;
public class InvokeRemoteNativeMethodImpl extends
PortableRemoteObject
implements RemoteNativeMethodInterface
{
    public InvokeRemoteNativeMethodImpl()throws
RemoteException
    {
        super();
    }
    public String[][] invokeRemoteNativeMethod() throws
java.rmi.RemoteException
    {
        HaveNativeMethods hnm = new HaveNativeMethods();
        String[][] resultSet = hnm.getDBTable();
        return resultSet;
    }
}
```

Table 8. Code for RMI-IIOP server

```
import javax.naming.InitialContext;
public class RemoteNativeMethodServer {
    public static void main(String[] args) {
        try{
            InvokeRemoteNativeMethodImpl irnmObj = new
InvokeRemoteNativeMethodImpl();
            Context ct = new InitialContext();
            ct.rebind("RemoteNativeMethod",irnmObj);
            System.out.println("Remote native method invocation
server by RMI/IIOP is ready!");
        }catch(Exception ex)
        {
            System.out.println("Error is "+ex.getMessage());
            return;
        }
    }
}
```

Table 9. Code for RMI-IIOP client

```
package remotenativemethod;
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.rmi.PortableRemoteObject;
public class RemoteNativeMethodClient {
    public static void main(String[] args) {
        Context ct;
        Object obj;
        RemoteNativeMethodInterface rnmi;
        try{
            ct = new InitialContext();
            obj = ct.lookup("RemoteNativeMethod");

rnmi=(RemoteNativeMethodInterface)PortableRemoteObject.n
arrow(obj,RemoteNativeMethodInterface.class);
```

```
String[][] result = rnmi.invokeRemoteNativeMethod();
for(int i=0;i<result.length;i++)
    if(result[i] != null)
```

```
System.out.println(result[i][0]+"\\t\\t"+result[i][1]+"\\t\\t"+result[i]
][2]);
        }catch(Exception ex)
        {
            System.out.println("Error is "+ex.getMessage());
            return;
        }
    }
}
```

Now let us compile all java sources and in particular using `rmic -iiop` to compile `InvokeRemoteNativeMethodImpl.class` in order to generate stub class for the client side and tie class for the server side.

Run the naming server by entering the next command line in DOS: `tnameserv -ORBInitialPort 1055`

Run the RMI-IIOP server by typing the next command line(place you in the `C:\rmi` directory and type all in only one command line):

```
java -classpath .
```

```
-Djava.naming.factory.initial=
com.sun.jndi.cosnaming.CNCtxFactory
-Djava.naming.provider.url=
iiop://localhost:1055
```

```
-Djava.library.path=.\remotenativemethod\
remotenativemethod.RemoteNativeMethodServer
```

Run the RMI-IIOP client by entering the next DOS command line:

```
java -classpath .
-Djava.naming.factory.initial=
com.sun.jndi.cosnaming.CNCtxFactory
-Djava.naming.provider.url=iiop://localhost:1055
-Djava.library.path=.\remotenativemethod\
```

```
remotenativemethod.RemoteNativeMethodClient
```

4. CONCLUSIONS

With the help of Sun's RMI-IIOP distributed computing technology and its JNI API, we can develop distributed software that integrate and recuperate platform-specific native libraries or native applications situated at another places in a network while it is achievable by using ORB cross-language feature. RMI-IIOP technology is easier to use and doses not need IDL to define interfaces in comparison with another ORB products (For example Borland VisiBroker ORB,or Orbix ORB).The marriage between RMI-IIOP and JNI make a distributed computing solution allowing the abandonment of IDL .

5. REFERENCES

- [1] R. Gordon, Essential JNI : Java Native Interface, New York: Prentice Hall RTR, 1st edition, 1998.
- [2] S. Liang, Java™ Native Interface: Programmer's Guide and Specification, US: Addison-Wesley Pub Co., ISBN: 0201325772, 1st Edition, 1999.
- [3] William A. Ruh, et al, IIOP Complete: Understanding CORBA and Middleware Interoperability, US: Addison-Wesley Pub Co., ASIN: 0201379252, 1st Edition, 1999.
- [4] R. Orfali, D. Harkey, Client/Server Programming with Java CORBA, US: John Wiley & Sons, ASIN: 047124578X, 2nd edition, 1998.
- [5] Qi Mnglong, Xiong Qianxin, et al, "A Distributed CORBA Event Service for Displaying a Data-Set Readed from a Database(in Chinese)", Journal of Wuhan University of Technology(Transportation Science & Engineering), Vol.26, No.3, June 2002, pp.321-323.
- [6] Qi Minglong, "Transformation of Any Database Table into a XML Document Using SAX and JDOM API(in Chinese)", Application Research of Computers, Vol.20 2003 supplementary issue, December 2003, pp.36-37.

Qi Minglong, born on September 1962 in Jiangsu province China, Associate Professor in School of Computer Science and Technology, Wuhan University of Technology, Ph.D. in visualization of scientific data achieved in Claude-Bernard University Lyon I (Lyon France) from 1984 to 1989, had worked from 1989 to 1999 for a French software development company as analyst-programmer and project manager. He had performed different post-doctoral research in the same period. He came back to Wuhan University of Technology in 2000. His principal interests are in distributed computing, mixed programming, Java 3D programming, and algorithms design and implementation in Java and C++. He has published about ten papers.

Guo Qingping is a Full Professor and a head of Parallel Processing Lab, dean of Computer Technology Institute in School of Computer Science and Technology, Wuhan University of Technology. He graduated from Wuhan University in 1968; from Huazhong University of Science and Technology in 1981 with specialty of wireless technology. He is a holder of K. C. Wong Award of UK Royal Society (1994); was a visiting scholar of City University and University of West Minster (1986~1988), Visiting Professor of the UK Royal Society (1994), Visiting Professor of Queen Mary and Westfield College, London University (1997~2000), Visiting Professor of National University of Singapore (2000), Visiting Professor of University Greenwich (2003). He is one of the DCABES international conference founder, was the chairman of DCABES 2001, co-chair of DCABES 2002, and will be the chairman of DCABES 2004. He has published two books, over 80 Journal papers, edited two DCABES Proceedings. His research interests are in distributed parallel processing, grid computing, network security and e-commerce.

A Technology to Improve Embedded-Linux Real-time Performance

He Keyou, Hung Minfeng

Department Of Computer Science, Wuhan University of Technology and Science
WuHan, Hubei Province, China

Email: mjohh@163.com Tel.: 027-86533613

ABSTRACT

Embedded system requires high real-time Capability. This paper analyses some key factors on interrupt-latency of Linux, and offer a technology to short cut the interrupt-latency, i.e. make use of PP to realize reenter for Linux kernel mode. The result is that real-time Capability of Linux has been improved to some extend. Some curves on off-interrupt time are also given to make the result more clear.

Keywords: Embedded system, Real-time system, Linux, Interrupt, PP

1. INTRODUCTION

With time goes by, embedded system and real-time system become more and more important. Though different embedded system has its different merit, the embedded Linux system is welcome widely. The following are some reasons:

- 1) Linux is used widely. The embedded Linux system could offer sufficient function of desktop computing, and it's source code is open, which is easy for us to mend it for some special capability.
- 2) Linux has supported most of the CMOS chips, including StrongARM, MIPS, PowerPC and so on in embedded system.
- 3) Linux is free, and its liberal spirit has absorbed more and more programmers make contribution to it.

But, Linux is an all-purpose OS, so it has some shortcomings as an embedded system. For example, Linux will prohibit interrupt call when some kernel threads are running, virtual file system will cause inconstant latency and so on. So it's necessary to do some optimization on Linux so as to improve its real time performance.

In this paper, we just analyze the mechanism of interruption in Linux, and consider it's infection to real time performance. In the end a solution will be given.

2. LINUX INTERRUPTION SUPERVISION

As we know, when external interrupt occur, OS will call interrupt handling routine and access kernel mode. To confirm true-running, the kernel mode is not reentrant, i.e., to confirm this part of key process do not be interrupted. So the system runs into off-interrupt mode. In this span of time, the OS send interrupt processes to correspond device drivers. Generally, the key part will run first and the rest will be push into the queue until the interrupt handling is over. For Linux, some data structures are set to tag the processes waiting for running:

```
Enum {
    TIMER_BH=0,  CONSOLE_BH,  TQUEUE_BH,  DIGI_BH,
    SERIAL_BH,  RISC8_BH,  SPECIALIX_BH,  ESP_BH,
    NET_BH,  SCSI_BH,  IMMEDIATE_BH,  KEYBOARD_BH,
    CYCLADES_BH,  CM206_BH,  JS_BH,
    MACSERIAL_BH,  ISICOM_BH};[1]
```

The different variable identifies different queue type, which shows different priorities.

```
extern unsigned long bh_active;
extern unsigned long bh_mask;
extern void(*bh_base[32])(void);
```

bh_base point array could contain 32 do-bottom-half procedures. bh_mask and bh_active show whether the correspond do-bottom-half procedure is exit or stimulated. if the No. N bit of bh_mask is 1 it means NO. N bit of bh_base array is an address of some do-bottom-half procedure. For example, if NO. N bit of bh_active is 1, it means the NO. N do-bottom-half procedure will be called by scheduler in suitable time. Those data structures are always renewed when OS initialize the exterior devices or call the interrupt handling routine. For example, when initialize the serial device in serial.c file, we use init_bh (SERIAL_BH, do_serial_bh) to assign a value to SERIAL_BH of bh_base [] array. Another example, in the serial device handling procedure, we use queue_task (&info->tqueue, &tq_serial) to push some tasks with low priority into ta_serial queue, which will be executed by a do-bottom-half procedure registered By bh_base, after the interrupt is over.

As we see, the interrupt handling routine has high efficiency, and satisfy many real-time needs. But the real-time performance of Linux is not good enough. Two reasons are given as following: First, Linux kernel mode is not reentrant. For Linux, system call is atomic operation. There are two kinds of process modes: client mode and kernel mode. When system is on kernel mode no other processes could interrupt it. If, in the mean time, there is a real-time process waiting for running, this real-time process will be hanging-up. So it will not run in time, which causes bad real-time performance. Second, latency of interrupt signals.

3. LINUX INTERRUPTION TEST

To visualize the interrupt latency, we test the interrupt latency of Linux. There are two types of interrupt: the synchronous and asynchronous. The asynchronous interrupt is more important for application and it is showed as Figure 1. The interrupt response time is the time slot between interrupt call and interrupt handling.

Interrupt response time is not a constant. It is related to OS and hardware component. Off-interrupt time is sum of interrupt assignment time and interrupt handling time. For Linux, _cli

`()/_sti()` is called to close or open interrupt-call. So the off-interrupt time is the time-slot between calling of `_cli()` and `_sti()`.

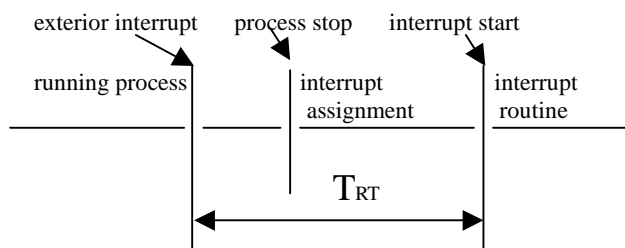


Figure1 asynchronous interrupt and interrupt response time^[2]

The interrupt latency application is used to calculate this time-slot. If we also record the type of off-interrupt, Figure3 and Figure4 will be got as following:

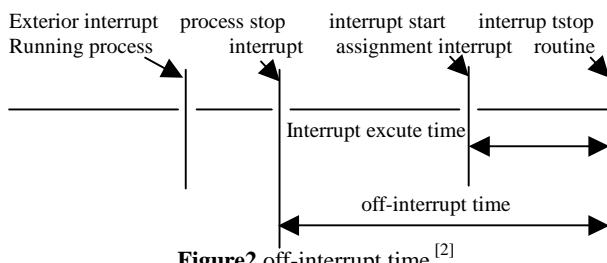


Figure2 off-interrupt time^[2]

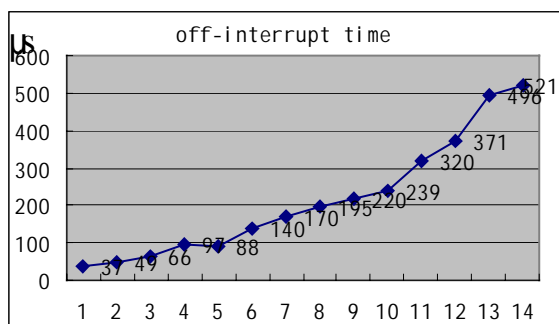


Figure3 off-interrupt time curve

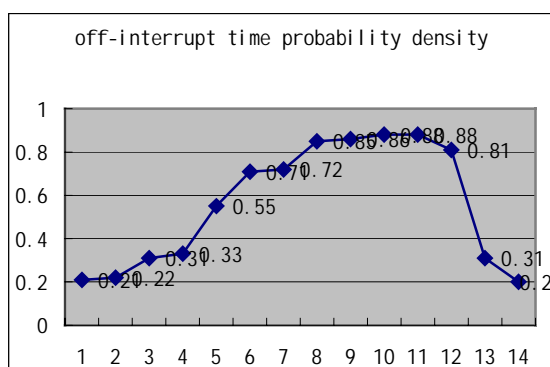


Figure4 off-interrupt time probability density curve

It takes us 5 hours to test the Linux, the result is given in Figure3. the application samples include disk copy loop and so on. We found the page switching cost most of time, which is surpassing 500μs.

4. MAKE USE OF PP TO ENHANCE REAL-TIME PERFORMANCE

As we know, essence of interrupt call is system call which is kernel mode^[3]. To shortcut the kernel latency, here, we make use of PP. When a system call get to PP and a real-time task is waiting for running at the same time, the system call will be replaced by real-time process, namely, the control power of CPU will transfer from kernel process to real-time process. A PP code will like the following: ^[3]

```
if(real-time-job-waiting) scheduler();
```

There is atomicity for system call that seems could not be splitted. However, in the reality, system call could be subdivided. Because there are some time gaps, namely PP, during the process of system call. ^[4] for example, the system will turn into dormant state when system call, read, is waiting for data from disk. And the CPU is free:

```
Sys-read-call;
Send request to disk;
Dormant until data is at disk;
Read a block from disk;
Read a block from disk;
Send Other request to disk;
here, a PP is insert and a short kernel latency has been got:
Sys-read-call;
Send request to disk;
do{
    if(real-time-job-waiting){
        save(&env_status);
        scheduler();
        resume(&env_status);
    }
    if(data is ready at disk){
        read a block from disk;
        ...
    }
} while (reading data is not over)
```

46 PP has been given in Linux2.2.16 patch:

PP	NUM
fs	14
kernel	9
memory	9
IPC	1
console driver	10
memory copy to user	3

Figure 5 PP distributing

In the reality, there are many PP in Linux, namely, PP has been used by Linux designers to cut down the Linux kernel latency.

To validate the affect of real-time performance, we test the Linux with 46 PP patch with the same application samples as above. For introducing the PP, the off-interrupt time should be redefined: the original off-interrupt time, T , is the time-slot between calling of `_cli()` and `_sti()`. t is the sum of

the PP witch has been made use of within a off-interrupt .Then the off-interrupt time, after introducing the PP, is $T - t$. The new off-interrupt time curve is following in Figure6:It take us 5 hours to test the Linux with PP patch, and the application for testing keep the same. The result is given in Figure6. We compare the off-interrupt time and give the result in Figure7 and found the off-interrupt time after introducing PP is generally less than that without introducing PP.

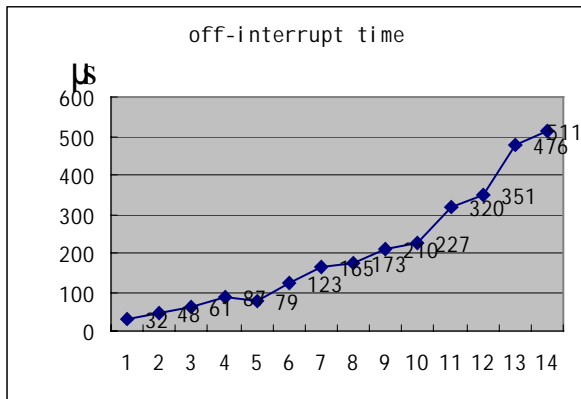


Figure6 off-interrupt time after introducing PP

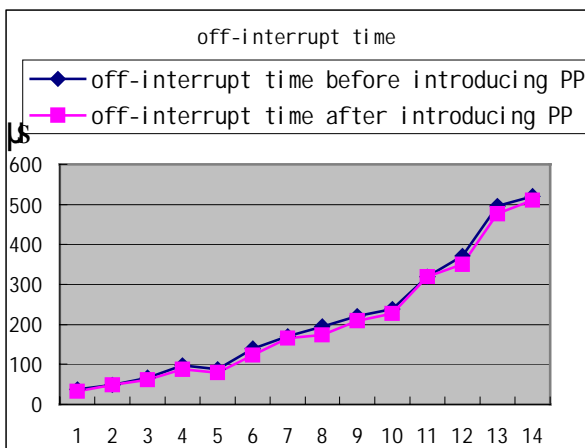


Figure7 off-interrupt time compare

- [3] YE Yimin ZHAO Huibin DI Zhengqiang. Embedded real-time OS. Observe and Control Tech. 2000(4): 6 ~ 8 in Chinese.
- [4] Tu Bibo Li Shengli A Technology of Enhancing the Real-time Capability of the Linux J.Huazhong Univ. of Sci.& Tech. Vol.29 No.12 Dec. 2001



He Keyou is an adjunct professor in School of Computer Science and Technology, Wuhan University of Technology. His research interests are in software engineering and now is devoted to embedded system.



Hung Minfeng is master graduate in School of Computer Science and Technology, Wuhan University of Technology. He is interested in embedded system development.

5. CONCLUSION

There are many ways to improve the real-time performance of embedded Linux. In this paper, we just talk about how to short cut the latency caused by not reentrant kernel mode including interrupt call and so on. The test result show that the real-time performance has been improved to some extends for making use of the technology of PP.

6. REFERENCES

- [1] YIN Ling, FEI Fei, WANG Xiaodong, YAO Tianfang Study of Transforming an Embedded Linux to a Real Time One. Computer Engineering Vol.27 No8 August 2001 in Chinese.
- [2] ZHAO Mingfu, LI Taifu CHEN Hongyan, HUXinyu. Analysis on Real time Performance of Linux Embeded System. ComputerEngineering Vo1.29 No18 October 2003 in Chinese.

Exploring the Initial Structures of Dynamic Markov Modeling for Chinese Text Compression

Ghim-Hwee Ong and Jun-Ping Ng
Department of Computer Science, School of Computing
National University of Singapore, Singapore

ABSTRACT

The paper explores the models for Dynamic Markov Compression (DMC) so that optimal compression on Chinese texts is obtained. DMC is found to be a very suitable method for the compression of Chinese texts when compared with some popular existing algorithms like LZW. By working with the special characteristics of GB2312 encoded Chinese texts, initial models which are able to exploit these characteristics to give good compression results are recognized.

Keywords: dynamic Markov modeling, text compression.

1. INTRODUCTION

Compression has been an active area of study in Computer Science, and notable effort has gone into developing newer and better compression techniques and algorithms. One reason for this is that compression reduces the amount of storage space a file occupies. This makes compression an important aspect of file archiving systems and document formats like Adobe Acrobat files or JPEG images. Compression also plays a big part in data communications. By reducing the amount of bits needed to represent a stream of data, compression effectively increases the throughput of a communications channel.

This paper aims at developing a method of Dynamic Markov Compression (DMC) to compress large Chinese text files which are encoded with the GB2312 encoding scheme [1]. This scheme is one of the most commonly used to encode Chinese text files today.

2. DYNAMIC MARKOV COMPRESSION

Dynamic Markov Compression (DMC) was proposed by Cormack and Horspool in 1987 [2]. It is a one-pass adaptive compression scheme, based on finite-state models. As with other statistical compression solutions, the compression process in DMC is partitioned into two steps – Modeling and Encoding [3, 4]. The model will track the characteristics of the input data, and pass a probability distribution of the data to the encoder.

The input data stream is processed symbol by symbol. As each symbol is processed, the model is updated and generated a new probability distribution for the encoder which generates the corresponding output data stream. The process is then repeated for the next symbol.

A Markov model is used as the modeler for DMC. Mathematically, Markov models are an abstraction of Markov chains. The National Institute of Standards and Technology (NIST) [5] carries a definition of Markov chains which is particularly apt for the context of this discussion – Markov chains are finite-state machines with probabilities for each transition, that is a probability of transitioning to the next state s_j given the current state s_i .

A possible Markov model for an alphabet of binary digits {0,1} is shown in the figure below. When the model is in state A, the probability of encountering '0' is 30% and the probability of encountering '1' is 70%. When a '0' occurs the model will then transit to state B, where the probability of encountering '0' and '1' is 10% and 90% respectively.

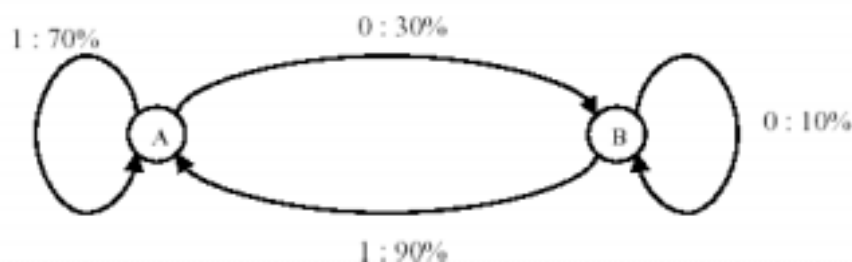


Figure1. Markov model for an alphabet of binary digits

In a practical implementation of DMC however, instead of storing the probabilities of a transition, frequency counts of the transitions are stored instead. That is to say if a out transition from A to B occurs x times due to the occurrence of '0', that transition will be linked with the count x . Similarly, if '1' is encountered y times, the count y will be stored with the transition for '1' from A to A. Storing frequency counts instead of probability estimates makes it easier to compute the probabilities only when required, instead of calculating them each time the model is updated.

Central to the performance of DMC is cloning. When the count of transitions out from a state reaches a certain threshold (the threshold is defined as two parameters MIN_CNT1 and MIN_CNT2), a new state is created out of the original one. In essence, the new node is to capture the context where the current outgoing transition is seen. The figure below provides an example. Before cloning, when the current state is D, it is not known which are the states (B or C) before A. To capture this information, a new state A' can be cloned from A so that it is now possible to know which state was visited before. Such information helps the model make a more accurate prediction

of the next occurring bit and allow the coder to compress the file more effectively.

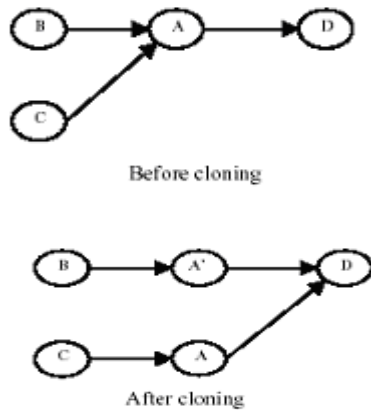


Figure2. The node state before and after cloning

A point to note is that it is not possible to keep on generating new states out of the model. A fixed amount of memory is allocated to the building of new states. When the memory available for new states runs out, Cormack et al. (1987) [2] suggested that the model built so far will be discarded and reset to the original one.

DMC uses an Arithmetic Coder to encode the input data given probabilities supplied by the Markov model [4, 6]. At any point in time, the probability distribution supplied by the Markov model will be the probability distribution of the current state of the model. Using the previous example, when the current state is state A, and the coder requests for a probability distribution, the probabilities for each symbol is calculated based on the frequency counts linked with each outgoing transition.

3. INITIAL STRUCTURES FOR DMC

It is known that Arithmetic Coding provides an optimal solution to the coding phase of compression. Modeling thus holds the key for improving compression results when applying DMC. The study into modeling revolves mainly around the design of an initial structure to be used by DMC. The cloning mechanism will then be responsible to grow this initial structure such that it represents the characteristics of the text to be compressed. In this paper, a class of 2-symbol based initial structures is proposed. As the name suggests, 2-symbol based structures use a single bit as the basic encoding unit ('0' and '1'), and the input data will then be read and processed bit by bit. There are several 2-symbol based initial structures that can be used with DMC:

- The **single state model** is the simplest model. It comprises of a single state with transitions leading out and back to itself.

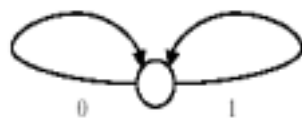


Figure3. The single state model

- The **linear model** consists of several states chained together in a straight line. The transitions leading out of a state lead to the state directly below it. The last state in the chain will have transitions leading back to the first state to form a cycle. A linear model of height n will thus contain a chain of n states before wrapping around.

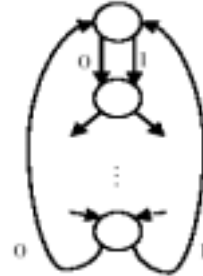


Figure4. The linear model

- The **tree model** resembles a binary tree where the transitions form the branches and the states form the nodes. Transitions out of the leaf nodes will lead back to the root. A tree of height n is defined to have $n-1$ transitions from the root node to the leaf nodes.

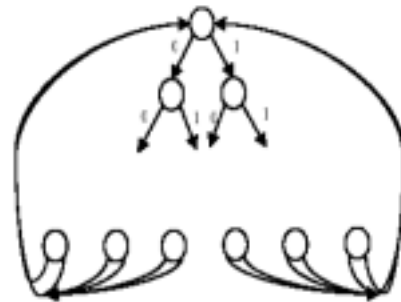


Figure5. The tree model

- The **braid model** is a generalization of the tree model where a bit sequence follow transitions from any top level node back to a unique top level node determined by the sequence.

4. RESULTS AND DISCUSSIONS

Other than the single state model, measurements of the accuracy of each of the models are done for different heights of the models. The figures shown in Table 1 are results of the optimal compression achievable by DMC for each of the six test files. These files are used as the sample files on which the compression algorithms are run:

- CHI001 (37KB) - Collection of science articles and news
- CHI002 (105KB) - Short Story 《边城》
- CHI003 (192KB) - Collection of short articles and commentary
- CHI004 (491KB) - Short Story 《灵山》
- CHI005 (559KB) - Collection of children tales
- CHI006 (1379KB) - Text of Chinese classic 《红楼梦》

Table 1. Compression achieved by different initial structures.

	CHI001	CHI002	CHI003	CHI004	CHI005	CHI006	Average
Single	84.48%	71.58%	77.06%	67.24%	62.78%	59.42%	70.43%
Tree							
2 levels	80.72%	66.39%	73.49%	63.11%	59.25%	56.84%	66.63%
4 levels	74.79%	62.37%	69.25%	60.01%	56.32%	54.08%	62.80%
7 levels	95.27%	87.30%	89.93%	79.51%	78.15%	73.00%	83.86%
8 levels	69.94%	59.07%	65.68%	57.30%	54.21%	52.50%	59.78%
12 levels	83.81%	73.01%	77.76%	68.01%	65.71%	62.33%	71.77%
14 levels	94.91%	85.49%	88.13%	78.95%	78.01%	73.89%	83.23%
16 levels	77.64%	63.50%	71.69%	63.58%	60.82%	58.76%	66.00%
Linear							
2 levels	79.34%	67.16%	74.08%	63.06%	59.41%	56.72%	66.63%
4 levels	76.08%	62.79%	70.02%	60.08%	56.80%	54.44%	63.37%
7 levels	94.01%	84.56%	88.39%	77.14%	75.96%	70.76%	81.80%
8 levels	72.89%	60.75%	67.62%	58.33%	55.18%	53.07%	61.31%
12 levels	80.17%	68.87%	74.64%	64.27%	61.00%	57.41%	67.73%
14 levels	90.83%	81.20%	83.97%	73.53%	72.36%	67.35%	78.21%
16 levels	73.39%	63.43%	70.03%	60.33%	58.22%	55.55%	63.49%
Braid							
1 row	83.95%	70.60%	77.70%	66.25%	62.64%	60.19%	70.22%
2 rows	80.93%	66.78%	74.05%	63.65%	60.14%	56.71%	67.04%
4 rows	76.19%	63.45%	70.28%	60.56%	57.19%	54.59%	63.71%
7 rows	93.40%	84.23%	87.24%	76.49%	75.29%	70.30%	81.16%
8 rows	70.58%	59.97%	66.45%	57.83%	54.90%	52.89%	60.44%
12 rows	87.30%	70.11%	74.51%	63.57%	61.11%	58.02%	69.10%
14 rows	111.47%	87.84%	91.26%	74.21%	70.89%	65.63%	83.55%
16 rows	93.46%	75.05%	79.08%	64.79%	59.71%	56.71%	71.47%

The percentages shown represent the ratio of the compressed file size over the original file size. The compression results are obtained by letting the compression program automatically determine the best values to use for the threshold values MIN_CNT1 and MIN_CNT2. These runs, and other following runs, are done on a Windows XP Laptop with an Intel Pentium III processor and 256MB of RAM. It should be noted however that the platform does not affect the compression results. Execution time may differ but the results are similar because the compression algorithm is deterministic and will produce the same output given the same input data.

It is found that for the Tree, Linear and Braid models, heights which are not factors of 16 do not produce good results. These three models perform best with 8 levels.

One explanation for the observation that the tree, linear and braid models perform best with 8 levels or rows is due to the way the GB2312 encoding table is arranged. Within a 2-byte GB2312 character, the first 4 bits of each byte is always at least 0xA. When a model of 8 levels is used, conceptually, this character of 16 bits is treated as 2 separate bytes. As the first 4 bits of each byte can only take values from 0xA to 0xF, such a separation reinforces this characteristic, and allows the model

to capture it distinctly. Specifically, the top half of the model which tracks the bits occurring in the first 4 bits of each byte will be able to more accurately predict the next successive bit as the transition counts for 0x0 to 0x9 are never incremented.

This explains also why reasonably good results are obtained with a model of 2, 4, 12 or 16 levels, and why these results are not as good as using 8 levels. For a model of 4 levels, the last 4 bits of each byte will make the characteristic less pronounced. Where the transition counts for 0x0 to 0x9 can be the minimum of 1 previously, these counts are now non-trivial and the model is thus unable to make full use of this characteristic. The same reasoning applies to a model of 2, 12 or 16 levels.

The table below shows the compression achieved by some of the common compression techniques. Compared with the effectiveness of other compression algorithms, the Tree, Linear and Braid models allow DMC (59.78%) to do better than compress (63.61%), Pack (76.78%) and SAC (80.74%). The single state model is able to outdo pack and SAC but is unable to do better than compress. It is thus seen that DMC when coupled with an appropriate model, will be able to compress Chinese text files better than the other existing algorithms.

Table 2. Compression Results of Various Approaches.

	CHI001	CHI002	CHI003	CHI004	CHI005	CHI006	Average
Compression Algorithms							
Compress	67.74%	61.16%	67.85%	64.07%	59.13%	61.72%	63.61%
Pack	78.60%	75.97%	77.83%	76.19%	76.34%	75.76%	76.78%
SAC	80.15%	83.40%	79.14%	82.46%	81.56%	77.74%	80.74%

As a quick summary, the table shows that DMC is able to perform better than the other algorithms for all files except CHI001. This may suggest that LZW is able to perform better

for smaller files (as the file sizes of CHI001, CHI002, and so on are in increasing order). However the difference between DMC and LZW in this case is not too significant. On the

whole, DMC will be able to give better compression results than the other algorithms listed.

5. CONCLUSION

DMC is suitable for compression Chinese text files. It is able to compress Chinese text files better than other commonly used compression algorithms such as Huffman coding or LZW. Existing compression packages currently utilizes algorithms like LZW. Since DMC is able to perform better than these algorithms, if they are incorporated into the compression packages instead, the packages may be able to perform better than now when used to compress Chinese text files.

The Tree, Linear and Braid models give the best compression ratios with 8 levels or rows. These 8 level models are able to capture distinctly the observation that the first 4 bits of each byte within a 2-byte GB2312 encoded character is always at least 0xA. Similarly, 4-level models are able to take advantage of this observation and produce good results too, whereas this relation is either lost, or diluted in the rest of the models.

6. REFERENCES

- [1] http://i18nwithvb.com/surrogate_ime/codepages/gbk.htm
- [2] Cormack, G.V. and Horspool, R. N. S. (1987), 'Data Compression Using Dynamic Markov Modeling', *The Computer Journal*, Vol 30 No 6, 1987 : 541 – 550
- [3] Timonthy, C. B., Cleary, J. G., Witten, I. H. (1987), 'Text Compression', Prentice Hall, New Jersey, 1990
- [4] Witten, I. H., Neal, Radform M., Cleary, J. G. (1987), 'Arithmetic Coding for Data Compression', *Communications of the ACM*, Vol 30 No 6 (1987) : 520 – 540
- [5] National Institute of Standards and Technology – Markov Chain, 2002
- [6] <http://www.nist.gov/dads/HTML/markovchain.html>
- [7] Vines, Phil and Zobel, Justin (1998), 'Compression Techniques for Chinese Text', *Software – Practice and Experience*, Vol 28 No 12, 1998 : 120-131

Research on Integration of Web Services and Workflow Modeling Technique

Shen Yuan , Chen Wen-bo , Yao Zhi-qiang
College of Computer Science, Beijing University of Technology
Beijing, 100022, China
Email: chen_wenbo@263.net Tel.: 010-67391745

ABSTRACT

Workflow process modeling is an important step in Workflow Management. This paper researches and implements a Web Workflow modeling system, which uses WSFL and design patterns to describe and realize the Workflow process model. In this system, new coming technology of Web Services was used to import existing process template during the process of modeling, in order to improve the quality and efficiency. So a new approach to model and realize the business process in the field of dynamic e-business is brought forward.

Keywords: Web Services; Workflow; WSFL; Modeling; Design Patterns.

1. INTRODUCTION

Some new trends of Workflow products and techniques are emerging now, such as embedding Workflow products in the enterprise application software kit, extending Workflow products to Web browser and etc [1]. However, overview today's Workflow techniques, there still are some evident limitations, especially lacking for a uniform standard of the Workflow modeling technology [2]. On the other hand, in the field of dynamic e-business, the research about Workflow business process between the enterprises has become the topic need to be solved urgently.

With the rapid growth of Internet technique and applications, the Web technique has stepped into the Web Services phases. Web Services differ from simple service such as search engine or online transaction. The main character of Web Services is offering the services to distribute application program so other programs can access it on Web. By using this technique, people can not only share data but also share applications.

Both the establishment of related standard and the support from enterprise have accelerated the development of Web Services. However, the function of single Web service is limited. What we really want is the business process cooperated by multi Web Services. Comparing with traditional Workflow process, this automatic business flow has not only some similarity but also a lot of new technique characters. The languages, which had been proposed to describe Web Services compositions, included Web Services Flow Language (WSFL) [3] from IBM, Xlang from Microsoft and Business Process Executing Language for Web Services (BPEL4WS) [4], the former two languages' combination, and etc.

This paper emphasize on the integration of traditional Workflow techniques and Web Services. We have designed

and realized a visual tool for Workflow modeling based on Web Services. At the same time we have abstracted and concluded a suit of feasible technical methods. On one side Web Services play a role of realizing the activities during the whole business process, on the other side they work as template suppliers during the modeling process. We build the business flow model through the visual work interface and use WSFL to define a XML expressing method to describe it. The method provides a uniform standard so that the application program on the server can invoke and manage the execution of Workflow process based on the model, and it makes the definition and realization of Workflow become natural and convenient.

2. KEY TECHNIQUES

2.1 Workflow Process Modeling

In 1996, Workflow Management Coalition (WfMC) defined the Workflow as: The computerized facilitation or automation of a business process, in whole or part [5]. It means that Workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal. The defined set of rules is the description of Workflow process, and it is very important to the Workflow management.

The definition of Workflow process actually is the formal description of the business process. It includes the definitions of all activities and information involved in the going process. The activity denotes a logic step in the Workflow, and the information consist of beginning conditions, ending conditions, involved activities and the rules navigated between them etc.

Because Workflow process and business rules often change, the key to develop and apply Workflow technique is presenting a new language or approach to define Workflow process, which suits with new circumstances.

2.2 Usability of Web Services to Workflow Modeling

Web Services are able to encapsulate the information, behavior, data representation and business flow based on uniform standard, no matter which systems and devices are used. The goal of Web Services is to enable seamless application integration over the network regardless of programming language or operating environment. The goal of Web Services Workflow is to enable the same type of seamless integration across business processes and transaction lifecycles that make use of many Web Services. At the same time, WSFL as a model description language based on XML applies a standard specification to Workflow modeling and it solves the deficiency of Workflow modeling technique in existence. Due to the flexibility in expressing an event by WSFL, we can decide Web Services work as a single activity or a sequence of activities during the business process. All these characteristics make Web Services very usable to

* This work is supported by the Beijing Municipal Education Commission.

Workflow modeling.

2.3 Web Services Flow Language

Web Services Flow Language (WSFL) is a new standard from IBM that addresses Workflow on two levels: (1) it takes a directed-graph model approach to defining and executing business processes; (2) it defines a public interface that allows business processes to issue themselves as Web Services.

WSFL is actually a tool to model activity graph, and it use XML representation, which both human and computer can understand easily. With the use of WSFL, a progressive Workflow engine according to the activities and control points is able to run through the whole business process. This is not a new concept, but what's important is that it can help to model business process spanning the boundary of technique and commerce, and it is just the limitation of most of traditional Workflow models.

3. WORKFLOW PROCESS MODELING BASED ON WEB SERVICES

3.1 WSFL Description of Flow Model Composition of Workflow process model

The activity graph of business process always consists of some nodes and links.

Nodes can be divided into task nodes and flag nodes by their function. Task nodes represent all types of activities and tasks that make up of the business process. Flag nodes include "begin node" and "end node". We use them to definitely express the beginning and the ending of the flow and regulate that there is only one "begin node" and one "end node" in a model. Furthermore begin node has no usher node and is the only entry while end node has no subsequence and is the only exit of the flow.

Links are the direct-route between usher node and subsequence node and can be divided into data links and control links. Control links dominate the transition from usher to subsequence, and always bind with a transition condition. When the transition condition is true, activity transfer occurs. After executing, an activity node can transmit data not only to its subsequence but also to other nodes. Data links can explicitly describe this relationship.

Formal description of Workflow process model

In our formal description of Workflow process model, the specification of activity is given as a seven-tuple:

$$\text{Activity}_i = \langle N_i, B_i, I_i, O_i, T_i, U_i, S_i \rangle$$

Where:

- N is the name of the activity. Any two activities' names are different and can be used to identify the activity:

$$\forall \text{Activity}_i, \text{Activity}_j, i \neq j: N_i \neq N_j$$

- B is a set of basic attributes, including activity type, icon and other information.

- I is a set of input parameters, and O is a set of output parameters.

- T is a set of conditions, defined as a three-tuple $\langle T_b, T_e, T_s \rangle$. T_b is the condition of starting the activity instance. T_e is the condition of ending the instance and T_s is the condition of changing activity's status.

- U is a set of roles involved in the process, such as users or managers.

- S is the current status of activity.

Control link is defined as a four-tuple:

$$\text{ControlLink} = \langle N_c, S_c, T_c, L \rangle$$

Where:

- N_c is the name of the control link.

- S_c is the mark of usher node.

- T_c is the mark of subsequence node.

- L is a Boolean expression of the transfer condition, which controls the movement of activities. The subsequence node is activated when L is true.

Data Link can also be denoted as a four-tuple:

$$\text{DataLink} = \langle N_d, S_d, T_d, D \rangle$$

Where:

- N_d is the name of data link.

- S_d is the mark of usher node.

- T_d is the mark of subsequence node.

- D is the set of data transferring between two activities.

Description of models using WSFL

WSFL can directly map the formal defined elements to XML marks. It supplies powerful data service to application program for controlling and managing the operation of Workflow.

According to the specifications of WSFL, tag **<activity>** is used for expressing the activity and attribute "name" denotes the name of the activity. Tag **<input>** and **<output>** denote the activity's input parameters and output parameters. Data link and control link can use tag **<dataLink>** and tag **<controlLink>** to express. The start point of link can be denoted by tag **<source>** and the end point of link can be denoted by tag **<target>**.

We can describe the flow model by integrating all these elements in a structural form. Here is a simple illustration of the structure of **<flowmodel>**:

```
<flowModel name=" " provider=" " >
  <activity name=" " >
    <input message=" " >
    <output message=" " >
    <implement
      ....
    </implement>
  </activity>
  <controlLink name=" " source=" " target=" "
    transitionCondition=" " />
  <dataLink name=" " source=" " target=" "
    <map sourceMessage=" " targetMessage=" " />
  </dataLink>
</flowModel>
```

3.2 Recursive composition in model

Nesting flow in recursive composition

In the process of defining business flow, some activities are called cell activities and others, which have sub-activities, are called nesting sub-flow. It is showed in figure 1.

We can enclose some related activities and use one group node to denote them. It can not only strengthen the expressive ability of business process graph but also use these process models to constitute a "template base" for later direct

reference. The template includes the topology information and transaction information of the process. The forming of the “template base” depends on the accumulation of knowledge and experience.

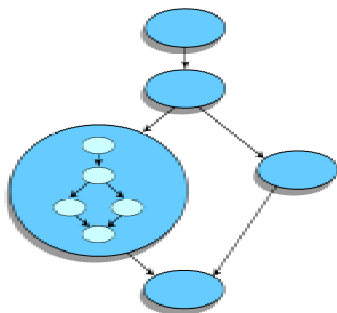


Figure1 Nesting flow in recursive composition

Maybe the business process is very complex. For this reason, it is not suitable to show the whole graph in one screen. The definition of “sub-flow” supports the top-down progressive modeling approach and makes cooperative modeling possible. The people who take charge in modeling do not need to know all details about the whole process because other experts can carry out the modeling of some special tasks. As a result, it ensures the quality of models and reduces the difficulty of modeling.

WSFL provides extensive support for the recursive composition of services: In WSFL, every Web Service composition can itself become a new Web Service, and can thus be used as a component of new compositions. The ability to do recursive composition of Web Services provides scalability to the language and support for top-down progressive refinement design as well as for bottom-up aggregation.

Application of "Composite Pattern" during the realization

To solve the problem mentioned above, we use the "Composite Pattern" [6] during the realization to offer a reusable program framework and to make the program more extensible. Composite pattern belongs to the structure pattern of object. It is often used to describe the relationship between whole and part, especially the object that has a tree structure. In this pattern there always are three roles: component, leaf and composite.

Component is the abstract role in the program, and it gives objects attended in the combination a standard interface and default operations. Leaf denotes the simple objects, which have no junior sub-objects. Composite denotes other kind of objects that have sub-objects.

Applying this pattern to concrete definition of objects, we can define cell activities and links as leaf and sub-graph nodes as composite. We also define an interface named graph component, which abstract common operation of all types objects and used by application program, and all other objects implement this interface. The relationship described by class diagram of UML is shown in figure 2.

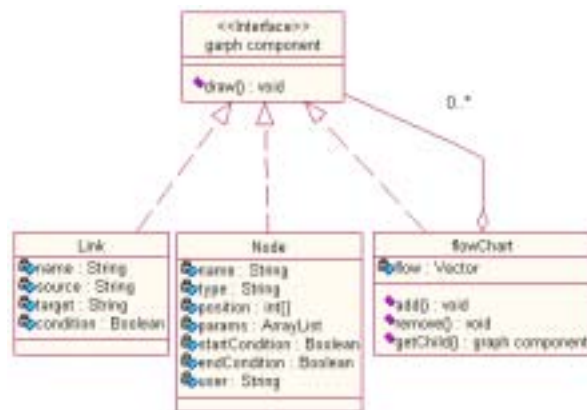


Figure 2 Application of “Composite Pattern”

3.3 UML Description in Validating the Model

As it is mentioned before, the people who take charge in modeling need to use accumulated knowledge and experience while fulfill some complex tasks. With the development of Web technique, modeling engineer can import mature defined flow model as template from Web sites and integrate existent models to define new business process. Furthermore, if they want, they can use Web Services to distribute their own models. After receiving template resources from Web site, application program need to check the correctness and validity. If the XML document is written obeying the rules listed in XML standard, it is correct. But the correct XML document does not equal to a valid document. Only if it also obeys a standard data format or a framework can we cognize it as a valid document. There are several approaches can define a document's data format. At present the most popular way is XML Schema. In real practice, we can use a visual tool to test it.

After designing the structure of the flow model, the design about activities and links involved in some numerical value information, and the information is related. For straight design Schema, we make use of Uniform Modeling Language (UML). According to assured mapping rules, we can use class diagram of UML to describe the XML document. It describes the relationship between data information exactly and straight and by using the modeling tool we designed, the information can be inputted.

The class diagram of UML illustrated the schema which was used to test the validity of Workflow model is showed in figure 3.

In order to individuate the Workflow activity graph, we use the “Stereotype”, one extensibility mechanism of UML, to map different types of elements in the XML document. As shown in figure 3, both class dataLink and class controlLink inherited from class linkType. The attribute “map” made references to the definition of class map.

The XML description of Workflow document mapping in terms of rules is as below:

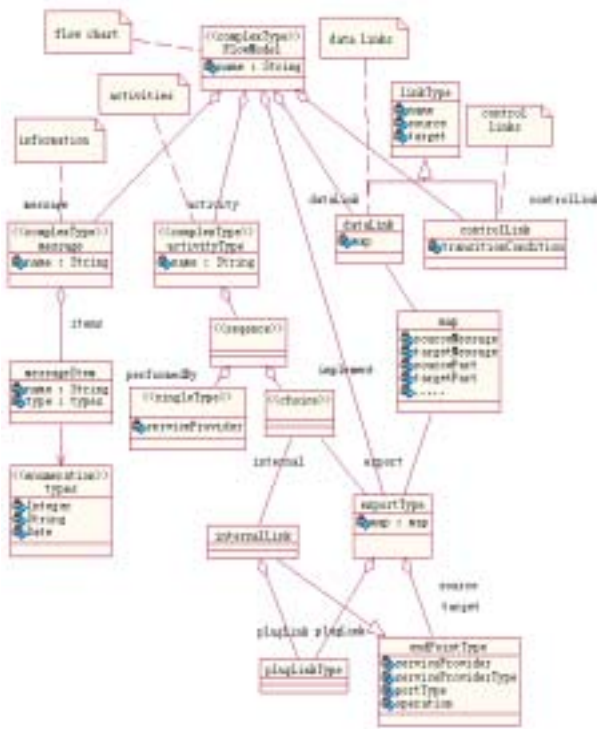


Figure 3 Using Class Diagram to Define Schema

```

<complexType name="linkType">
  <attribute name="name" type="String">
  <attribute name="source" type="String">
  <attribute name="target" type="String">
</complexType>
<complexType name="controlLink">
  <complexContent>
    <extension base="linkType">
      <attribute name="transitionCondition" type="String">
    </extension>
  </complexContent>
</complexType>
<complexType name="dataLink">
  <complexContent>
    <extension base="linkType">
      <element ref="map">
    </extension>
  </complexContent>
</complexType>

```

Aggregation relation was used here. Class plugLinkType aggregate to class exportType. The corresponding XML document likes:

```

<complexType name="exportType">
  <sequence>
    <element name="source" type="endPointType">
    <element name="target" type="endPointType">
    <element ref="map">
    <element name="plugLink" type="plugLinkType">
  </sequence>
</complexType>

```

The recursive composition mentioned above was also embodied in this graph. Several classes, including class activityType and class exportType, aggregated to class

flowModel. Class activityType contained stereotype <<choice>>, which means including several optional items. By this means, class activityType had the possibility to import an object of class exportType. Then the recursive composition was realized by WSFL. In face, the strict and standard description of WSFL was achieved under the cooperation of modeling tools in an interaction way.

```

<complexType name="activityType">
  <complexContent>
    <sequence>
      <element name="performedBy">
        <attribute name="serviceProvider" type="String"/>
      </element>
      <element name="implement">
        <complexType>
          <choice>
            <element name="internal" type="internalLink"/>
            <element name="export" type="exportType"/>
          </choice>
        </complexType>
      </element>
    </sequence>
    <attribute name="name" type="String"/>
  </complexContent>
</complexType>

```

3.4 An Example about Using Web Services

While we using modeling tool, we can log in from Web browser and input our username and password. After passing the id check, we entered into the modeling Web page. Web server found the needed Web Services supplier registered on it and bind the Web Services. The wanted template was returned to the client browser. The process shown in sequence diagram of UML is as following:

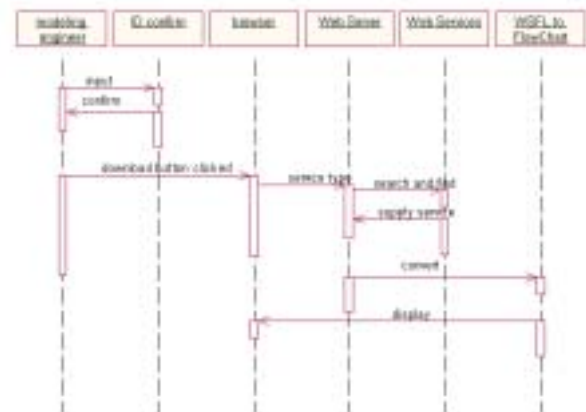


Figure 4 Sequence diagram about Importing Template

The core code on application client for binding to the Web Services is:

```

try {
  String wsdlUrl=
    "http://cscwserver:7001/web-services/
    exporttest?WSDL";
  ProviderBean service =
    new ProviderBean_Impl(wsdlUrl);
  //connect to Web Services supplier
  ProviderBeanPort port =
    service.getProviderBeanPort();
  PrintWriter out = response.getWriter();
}

```


A Trust Management Framework Suitable for Web Services Security *

AI Ping, MAO Ying-Chi
 School of Computer and Information Engineering, Hohai University
 Nanjing, Jiangsu 210098, China
 Email: aip@hhu.edu.cn Tel.: 86-25-83786520

ABSTRACT

It is very important to ensure services security for the web services security system. At present, there exist many problems about non-integrity of security information in the web services system. So in this paper, we discuss some basic concepts about direct, linear recommendation and peer-to-peer trust relationship among services and self-contained trust domain, furthermore establish an open, flexible trust management framework suitable for web services security.

Keywords: web services security, security grade, the trust management framework.

1. INTRODUCTION

With the growth of web services technology, reconstruction and running mode of application software system had been changed thoroughly. Openness of web services brought many problems about security, which resulted in invalidation of techniques and methods of security based on tradition application (such as access control list (ACL), X.509, PGP and so on) [1]. It's primary reason that the architecture for security authentication and certification is unsuited for non-integrity of security information in open system. Therefore, M. Blaze et al used concept of "trust management" in 1996 firstly. Adul-Rahman et al divided concept of trust into trust content and trust grade and gave a mathematical model for trust evaluation based on subjectivity.

For discussion trust in the field of information security, trust should be defined accurately. But, from the 90s of last century to today, we cannot form consistent understanding about trust. A. Jøsang discussed many definitions of trust in the end of last century [2].

A accepted definition of trust from D. Gambetta (1990) is: "trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent will perform a particular action, both before [we] can monitor such action (or independently of his capacity of ever to be able to monitor it) and in a context in which it affects [our] own action" [5].

The definition brought far-reaching influence on subsequent research. Its cores are: trust is subjective, actions of trust cannot be monitored and the level of trust depends on how our own actions are in turn affected by the agent's actions. In other words, studying on trust mainly focuses on trust expression, trust measurement and security grade evaluation, and the key is the security grade evaluation.

M. Blaze et al defined that trust management is to adopt a consistent method to describe and explain security policy, security credential and also be used to directly authorize to perform the key security operation on trust relationship [6]. In addition, they presented a trust management model whose core is a trust management engine (TME). Figure 1 illustrates the trust management model. The function of TME is to make a judgment whether trust is coincident with the security policy or not, but the algorithms are quite difficult in mathematics [9] [16]-[18]. Some known trust management systems are all based on that model such as PolicyMaker [7], KeyNote [8] and REFEREE [9].

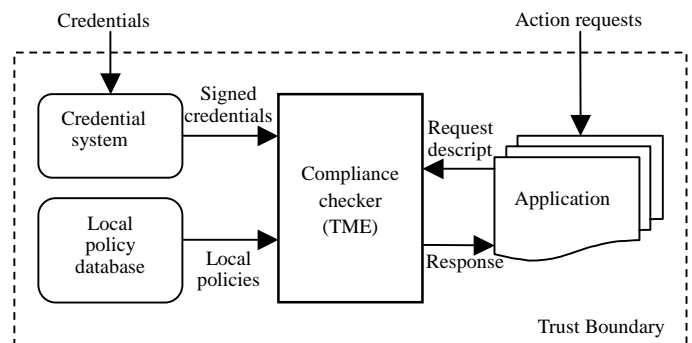


Fig.1 Trust management model

D. Povey put forward an idea of the subjective trust model. He thought trust management is to obtain, evaluate and implement trusting intention [14]. P. Hermann et al presented the concept of "trust-adapted enforcement of security policy" [15].

M. Blaze et al thought that trust can be described and managed accurately and rationally, while D. Gambetta, A. Adul-Rahman et al considered that trust is an experience and non-rational [5][10]. Therefore, trust not only includes concrete contents, but also has different grades division. Furthermore, several security grade evaluation models are given based on the above-mentioned opinions.

Beth's security grade evaluation model introduced experiential concepts to express and measure trust relationship, and presented the security grade deduction and comprehensive computation formula. Besides, A.Jøsang et al imported the concepts of evidence space and opinion space to describe and measure trust relationship, and further offered a suit of subjective logic operators to deduce and comprehensive computation security grade [19]-[21].

After all-round analysis existing research achievements, XU Feng and LÜ Jian [3] thought there are many issues in the existing trust management systems. Firstly, they only consider services not the security in the services requester. Secondly, it is too absolute for security measurement and too accurate for validation the policy consistency, which results in the poor adaptation. Thirdly, it is equivocal to express and measure

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2001AA113170 ; the National Grand Fundamental Research 973 Program of China under Grant No.2002CB312002; the Foundation of Nature Science and Hi-Technology of Jiangsu Province of China under Grant No. BK.2001016.

trust relationship with event probability based on hypothetical probability distribution, which could not reduce the bad effect on security grade evaluation from malign recommendation. Finally, there are many flexibility mistakes.

All in all, as far as solve the problems on web services security are concerned, the existing research results in the trust management is quite difficult to put into practice. Therefore, based on the existing results, this paper presented a trust management framework suitable for web services security.

2. TRUST AND SECURITY CHARACTERISTIC OF WEB SERVICES

2.1 Security characteristics

Web applications based on web services are multi-category services composition system in the networks, and services can dynamically be composed by other services. So web services security has many new characteristics and requirements, which are as follows:

- (1) Objects complexity involved in security. The objects involved in the web applications system security contain services providers and services register, as well as users and the system owners.
- (2) Security information non-integrality. The essence of services is open, so it is quite impossible for services providers and users to utterly control the other's security information.
- (3) Security measurement difficulty. Open and dynamical web services environment brings many indeterminate factors to influence the system security and result in greater difficulties in evaluation system security.
- (4) Changing security requirements. Web services environment is greatly dynamical and changeable, as well as applications based on web services often need to dynamically adjust, therefore the security measures must be adapted the real-time changes.

Because services encapsulate information and information processing, it is very important to ensure the services security for the web services security system and trust management is one of the most efficient approaches to implement the based on services security guarantee.

2.2 Trust based on web services environment

Under the environment of web services, applications dynamically request services and implement the function by exchanging messages. So applications, as services requesters, could not accurately know about the security information of the required services. Moreover, a service encapsulates information and information processing, runs in its own environment and maybe need to request other services to implement its own function. Obviously, the traditional access control techniques are unavailable.

As we all known, inter-trust is the basis for collaboration among persons. As a whole, such trust is only the subjective sense, while the trust grade judgment is related to the contents and goal of collaboration. For example, *A* wants to buy things from *B*. From *A*'s respective, the *B* trustiness embodies that *B* can provide the required goods, sound price, high quality, timely delivery and good after services, while from *B*'s respective, the *A* trustiness embodies that *A* can pay the price of goods on schedule, have skills to correctly use the bought goods. If *A* thinks *B* is unbelievable, *A* could not buy things

from *B* and search other providers or require *B* to reduce the price. Of course, if *B* thinks *A* is unbelievable or *A* is not a good buyer, *B* can obviously refuse to provide services or require a higher price from *A*. On the other hand, if *B* is not the producer, only sellers, while *C* is the producer, the products quality guarantee of *B* must be based on the *C* trustiness. So the trust grade between *C* and *B* would influence the *B*'s quality guarantee to *A*.

The situation of web services is analogous to that of the above. Now there are three main elements in the web services architecture; they are "Service Provider, *SP*", "Service Requester, *SR*", "Service Registry" or "Service Agency, *SA*". Firstly, *SR* registers in the *SA*; then *SP* searches the appropriate *SR* through *SA*; finally *SP* binds the services provided by *SR* and obtains services. During the process, the trust problems are involved with *SP*, *SR* and *SA*.

From *SR*'s respective, *SR* needs to find a believable *SA* and search a believable *SP* through *SA*. As far as *SP* is concerned, *SP* needs to find a believable *SA* to register services and expect *SA* not to commend itself to an unbelievable *SR*, while *SA* requires that *SR* and *SA* are all believable. Analogous to the collaboration relationship among persons, the trust relationship among *SP*, *SR* and *SA* is absolutely subjective and pertinent to the contents and goals of every service. In addition, every role may require different security grade in every service. Therefore, the requirements of trust are dynamic and gradational.

There are many multi-aspect factors to influence the trust relationship among *SR*, *SP* and *SA*, in which security and quality of service are two of the most important factors. This paper only studies the trust problems on the security aspect, and the basic principles are also appropriate to deal with the trust problems on the quality of service aspect. So the following sections are limited in the security aspect.

SP, *SR* and *SA* are all the participants to guarantee security. Under the open network environment, the trust grade among them and credibility must be authenticated by the third party, which is called trust management systems.

According to the requirements of trust management, the function of trust management systems is to evaluate, register and maintain the security grade of web services subjects and authenticate trust in accordance with the requirements of subjects. The function implementation is dependent on trust denotation, trust policy expression, security grade evaluation and the matching algorithms for trust policy. In the first section of this paper has introduced many results in those aspects. The latest and valuable results can be found in <http://www-900.ibm.com/developerWorks/cn/webservices/>, <http://www.ibm.com/developerworks/library/> and the reference papers [22]-[24] [4]. Based on the analysis and study the achievements, we have presented a trust management framework suitable for web services security in the next section.

3. TRUST MANAGEMENT FRAMEWORKS IN WEB SERVICES ENVIRONMENT

3.1 Basic ideas

We give the following constraints in order to simply the problem description.

- (1) Trust management is only limited to services and the security of services transaction gradation, while it doesn't touch upon the details of services running security, which have discussed in the security specification of web services. Security grade reflects the all-round behavior of security details.
- (2) We only focus on the trust management framework for web services, while pay little attention to the technological details for implementation all parts in the framework. In the previous section, other results have provided many technological selections.
- (3) All of the concerned objects of web services are satisfied with the accepted technological specifications.

First, some denotations will be given. *SP*, *SR* and *SA* denote the Service Provider set, the Service Requester set, and Service Registry or Service Agency set respectively. In addition, *TM* and *S* denote Trust Management set and web services trust subject set respectively.

So *S* can be expressed as:
 $S = \{SA, SP, SR, TM\}$.

There exist inter-trust requirements between *SA*, *SP*, *SR* and *TM*, which can change with the contents and goals of services. That is the trust relationship *TR*.

$TR(S)$ can denote the inter-trust relationship between the elements in *S*:

$$TR(s_1, s_2, \dots, s_i \dots s_n), s_i \in S, i=1, 2, \dots, n, n>0.$$

Obviously,

$n=1$ indicates *TR* is the self-trust relationship, that is subject self-trust.

$n=2$ indicates *TR* is the direct trust relationship.

$n>2$ indicates *TR* is multiple complex trust relationship or recommendatory trust relationship. When *n* is big enough, *TR* maybe lose the application values.

But if :

$$TR(s_1, K, s_n) = \sum_{i=1}^{n-1} TR(s_i, s_{i+1})$$

Here, \sum is the connection operator of trust relationship. $n>2$, indicates the linear transitivity of trust relationship called linear recommendatory trust relationship.

In practice, the values of *n* need to be properly limited to improve the efficiency and reduce the complexity of trust relationship.

The one and only value of $TR(S)$ from the real number between 0 and 1 can define the trust relationship grade. The "existing $TR(S)$ " is the other denotation, which indicates the trust relationship is acceptable, while "non-existing $TR(S)$ " indicates the trust relationship cannot be established.

In view of the trust subjectivity, in general,

$$TR(s_1, s_2) \neq TR(s_2, s_1).$$

But if $TR(s_1, s_2) = TR(s_2, s_1)$,

then *TR* is called peer-to-peer trust relationship.

It is called trust policy that the requirements or commitments which the elements of *S* require. The policy can be expressed with trust assertion and adjusted based on the requirements.

On the assumption that

$$tm \in TM,$$

if there exists

$$TR(s_i, tm), s_i \in S \quad (\{s_i\} \cap TM = \emptyset), i=1, 2, \dots, n,$$

then $TD = \{\{s_i\}, tm\}, i=1, 2, \dots, n$,

is called trust domain

and *n* is called the dimension of *TD*.

If there is no $TR(s_i, tm_0) \quad tm_0 \in TM$,

then *TD* is close,

otherwise *TD* is not close.

If *TD* is close,

then $TR(TM)$ is the trust relationship between the trust domains.

In the same way, based on the above ideas, we can define the direct trust relationship and recommendatory trust relationship between the trust domains. So it is very useful to simplify the recommendatory trust relationship between non-elements of *TM* with the trust relationship between trust domains.

If *TD* is close and *SP* registers and only registers in the *SA* of the same *TD*,

then *TD* is call self-contained.

3.2 Construction of trust management framework

In the web services trust management framework, services are considered as basic objects, which require security management. Base on web services application systems is the dynamic composition of the required services, whose security must be guaranteed by the required services and application. From the respective of services security grade, in the process of finding and binding services, the application security is determined with service security grade. Therefore, trust management framework must be up to this characteristic.

Figure 2 illustrates a new architecture of web services appending trust management.

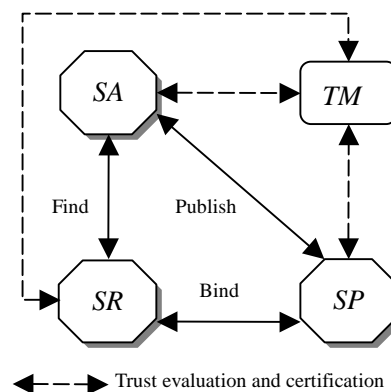


Fig. 2 A new architecture of Web Services

In the figure 2, in the process of publishing, finding and binding services, *SA*, *SP* and *SR* all require to communicate with *TM* and obtain the relevant trust authentication and permission. In order to implement trust authentication and permission, they also need to provide the relevant material to *TM* and evaluate the security grade of the relevant subjects.

Based on the above architecture, we establish a trust management framework, which consists of many elements. They are as follows:

$$TMF = \{Subject, Trust-Token, Operation\}.$$

Subject denotes all kinds of participant objects that require trust to be evaluated and authenticated. Usually subjects

represent all sorts of services.

Trust-Token is a token used to record information relevant to trust, which include the subject's identity (service requester and service provider), trust grade, trust administrator's identity, trust policy, services authorization and license. Trust-Token is an extensible table and basic information package for trust grade authentication between the service provider and requester, which exists in the process of services transaction. Token can be described by extension the SOAP messages. IBM et al have given explicit and concrete discussion to the similar description methods.

Operation contains various processing functions in terms of trust token.

It can improve the adaptation of framework by extension and modification the Trust-Token and Operation. So the framework is flexible.

According to the above framework, in a self-contained trust domain, a transaction process of finding services are as follows:

- *SR applies a Trust-Token to TM for access to SA, and then TM evaluates the trust for SR and decides whether issues a Trust-Token or not to SR.*
- *SR holds the Trust-Token to access SA and find the required SP. SA can make a decision whether provides services to SR or not based on Trust-Token and add the relevant information in the Trust-Token.*
- *SR holds the Trust-Token and applies to TM for access to SP. Then TM makes a trust evaluation based on the SP's entrustment and returns the results to SR by appending the relevant information in the Trust-Token.*
- *SR holds the Trust-Token authenticated by TM and applies to SP for services.*
- *SP makes a decision whether provides services or not based on Trust-Token and appending information in the Trust-Token. If SP accepts the service request, it would perform the action of service binding.*

In the above process, *SR* also makes a decision to return to the previous operation, go forward to the next step and terminate operation according to its own trust policy. When *SR* returns to the previous operation, the authentication of the previous operation in the Trust-Token is still valid.

After finishing all of operation, whether operations are completed or terminated, *SR* holds Trust-Token to apply to *TM* for withdrawing the Trust-Token. At that time, *TM* would record the content of Trust-Token in the Database to evaluate trust.

Other web services operations, such as services publication, are similar to the above operations.

3.3 Trust domain and entrusting

In the trust management, Subject *s1* can entrust Subject *s2* to evaluate trust. At that situation, $TR(s1, s2)$ is called entrusting relationship.

If *s1* obtains the trust information about subject *s3* through *s2*, and $TR(s1, s3)$ is indirect trust relationship,

as well as $TR(s1, s2, s3)$ is the recommendatory trust relationship,

then there must exist entrusting relationship between *s1*, *s2* and *s3*.

Under the open web services environment, it is impossible for application systems to be operated under the management of single trust management system. So it is obvious that results in the trust management problems cross trust domain.

Suppose trust domain (*TD*) is self-contained,

$TD1, TD2 \subseteq TD$,

$TD1 \cap TD2 = \emptyset$,

if

$s1 \in TD1$,

$tm1 \in TD1$,

certainly, $TR(s1, tm1)$ is an entrusting relationship

and if

$s2 \in TD2$,

$tm2 \in TD2$,

in a similar way, $TR(s2, tm2)$ is also an entrusting relationship,

as well as if $TR(tm1, tm2)$ exists and is peer-to-peer trust relationship,

then *s1* and *s2* can establish the recommendatory trust relationship by *tm1* and *tm2* to implement the service transactions.

Due to withdrawal Trust-Token only in the *tm1* or in the *tm2*, in order to maintain $TR(tm1, tm2)$, *tm1* and *tm2* must withdraw information relevant to Trust-Token to prevent $TR(tm1, tm2)$ from becoming non peer-to-peer trust relationship.

In the reality, the same services maybe register in the several registry or entrust many trust management services to administrate the service, which may do no harm to the service security. However, many trust entrustments can bring the differences in the security grade because of the differences of trust policy and evaluation methods, which can make it difficult to establish the trust relationship. Therefore, it is quite favorable to implement trust management by establishing many self-contained trust domains and peer-to-peer trust relationship among trust domains in the web services environment.

4. CONCLUSIONS AND DISCUSSIONS

Based on the previous research results, this paper presents one basic idea of web services trust management and illuminates about elementary concepts and framework in order to solve the security problems of web services.

Because Web Services is a service-oriented architecture, trust management not only solves the security problems, but also is a feasible approach to solve the quality of service problems. However, due to the trust subjectivity and trust relationship subjectivity, although much research work has done in the aspect of trust description, measurement and algorithms, there are still many difficult problems to be solved [3]. We want to establish more practical technological systems by the means of abstraction the problems from application requirements. In this paper, we only do pilot study. The further research topics are as follows:

- (1) The mathematical description of the trust and trust relationship as well as deduction and testifying the relationship operation in the web services environment.
- (2) Denotation and operation details of Trust-Token and representations and specifications of Trust-Token extension in the SOAP message.

- (3) Multiple, complex trust relationship simplification.
- (4) Management associated with services registry.
- (5) Systematically merging trust management and security, the quality of service management.

5. REFERENCES

- [1] Khare, R., Rifkin, A. Trust management on World Wide Web. *World Wide Web Journal*, 1997, 2(3), pp.77-112.
- [2] Jøsang, A., The right type of trust for distributed systems. In: Meadows, C., ed. *Proceedings of the 1996 New Security Paradigms Workshop*. Lake Arrowhead, CA: ACM Press, 1996.
- [3] XU Feng, LÜ Jian, Research and Development of Trust Management in Web Security. *Journal of Software*. Vol.13, No.11,2002. pp. 2057-2063
- [4] A.Jøsang, I.G.Pedersen and D.Povey, PKI seeks a trusting relationship. In *Proceedings of the Fifth Australasian Conference on Information Security and Privacy (ACISP 2000)*, Brisbane, Australia. Springer-Verla. July 2000.
- [5] D. Gambetta, Can We Trust Trust?. In: *Trust: Making and Breaking Cooperative Relations*, Gambetta, D (ed.). Basil Blackwell. Oxford, 1990, pp. 213-237.
- [6] Blaze, M., Feigenbaum, J., Ioannidis, J., et al, The role of trust management in distributed systems security. In: *Secure Internet Programming: Issues for Mobile and Distributed Objects*. Berlin: Springer-Verlag, 1999, pp. 185-210.
- [7] Blaze, M., Feigenbaum, J., Lacy, J., Decentralized trust management. In: Dale, J., Dinolt, G., eds. *Proceedings of the 17th Symposium on Security and Privacy*. Oakland, CA: IEEE Computer Society Press, 1996, pp. 164-173.
- [8] Blaze, M., Feigenbaum, J., Keromytis, A.D., Keynote: trust management for public-key infrastructures. In: Christianson, B., Crispo, B., William, S., et al., eds. *Cambridge 1998 Security Protocols International Workshop*. Berlin: Springer-Verlag, 1999, pp. 59-63.
- [9] Chu, Y.-H., Feigenbaum, J., LaMacchia, B., et al. REFERENCE: trust management for Web applications. *World Wide Web Journal*, 1997,2(2),pp.127-139.
- [10] Abdul-Rahman, A., Hailes, S. A distributed trust model. In: *Proceedings of the 1997 New Security Paradigms Workshop*. Cumbria, UK: ACM Press, 1998. pp. 48-60. <http://www.ib.hu-berlin.de/~kuhlen/VERT01/abdul-rahman-trust-model1997.pdf>.
- [11] Abdul-Rahman, A., Hailes, S., Using recommendations for managing trust in distributed systems. In: *Proceedings of the IEEE Malaysia International Conference on Communication'97 (MICC'97)*. Kuala Lumpur: IEEE Press, 1997. <http://citeseer.nj.nec.com/360414.html>.
- [12] Yahalom, R., Klein, B., Beth, T., Trust relationships in secure systems — a distributed authentication perspective. In: *Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy*. IEEE Press, 1993, pp. 50-164.
- [13] Beth, T., Borcherding, M., Klein, B., Valuation of trust in open network. In: Gollmann, D., ed. *Proceedings of the European Symposium on Research in Security (ESORICS)*. Brighton: Springer-Verlag, 1994, pp. 3-18.
- [14] Povey, D., Developing electronic trust policies using a risk management model. In: *Proceedings of the 1999 CQRE Congress*. 1999. 1-16. <http://security.dstc.edu.au/staff/povey/papers/CQRE/123.pdf>.
- [15] Herrmann, P., Krumm, H., Trust-Adapted enforcement of security policies in distributed component-structured applications. In: *Proceedings of the 6th IEEE Symposium on Computers and Communications*. Hammamet: IEEE Computer Society Press, 2001, pp. 2-8. <http://www.computer.org/proceedings/iscc/1177/11770002abs.htm>.
- [16] Blaze, M., Feigenbaum, J., Strauss, M., Compliance Checking in the PolicyMaker Trust Management System. In: Hirschfeld, R., ed. *Proceedings of the Financial Cryptography'98. Lecture Notes in Computer Science 1465*, Berlin: Springer-Verlag, 1998, pp. 254-274.
- [17] Blaze, M., Ioannidis, J., Keromytis, A., Trust management for IPSec. In: *Proceedings of the Internet Society Symposium on Network and Distributed Systems Security (SNDSS 2001)*. 2001. pp. 139-151. <http://www.cis.upenn.edu/~strongman/papers/tmipsec.PDF>.
- [18] Blaze, M., Feigenbaum, J., Ioannidis, J., et al, The KeyNote trust management system version 2. Internet RFC 2704, 1999.
- [19] Jøsang, A., A model for trust in security systems. In: *Proceedings of the 2nd Nordic Workshop on Secure Computer Systems*. 1997. <http://security.dstc.edu.au/staff/ajosang/papers.html>.
- [20] Jøsang, A., Knapkog, S.J., A metric for trusted systems. *Global IT Security*. Wien: Austrian Computer Society, 1998, pp. 541-549.
- [21] Jøsang, A., A Subjective Metric of Authentication. In: Quisquater, J., ed. *Proceedings of the ESORICS'98*. Louvain-la-Neuve.: Springer Verlag, 1998, pp. 329-344.
- [22] Stephen Weeks, Understanding Trust Management Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'01)*. IEEE 2001, pp.94-105
- [23] Trevor Jim, SD3: a trust management system with certified evaluation. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P'01)* IEEE 2001, pp.106-115
- [24] Ninghui Li, John C. Mitchell, William H. Winsborough, Design of a Role-based Trust-management Framework. *Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P'02)*. IEEE 2002, pp. 114-130



AI Ping is a Ph.D. and a full professor, dean of the Computer and Information Technology Institute in the School of Computer and Information Engineering, Hohai University. He has undertaken several research projects supported by the National High-Tech Research and Development Plan of China; the National Grand Fundamental Research 973 Program of China; the Foundation

of Nature Science and Hi-Technology of Jiangsu Province of China; the Foundation of Science and Technology of the Ministry of Water Resources of China, and he has published a book, over 30 papers in the journals and the international conference proceedings. His research interests are in architecture of computer application system in specific domain, intelligent data processing, software component and internet-ware.



MAO Ying-Chi is a Ph. D. candidate of the Department of Computer Science, Nanjing University, studying the Grid Computing, P2P Computing and Web Services. She is undertaking the research projects, including the Organization, Schedule and Management of Servers supported by the National High-Tech Research and Development Plan of China, and Research on the Flexible Composition

of Software Components in the Water Resources Allocation Applications supported by the National High-Tech Research and Development Plan of China. She was rewarded the master's degree of Computer Science at Hohai University in 2003. As a student member of IEEE, she has published several papers in the key Chinese journals and the international conference proceedings.

Research of Intelligent Search Engine Technology Facing Electronic Commerce*

Tong XiaoJun Wang Zhu
Computer Department, HarBin University of Technology (Weihai),
Weihai City, ShanDong Province 264209, China
E-mail: tong_xiaojun@163.com Tel: 13061181039

ABSTRACT

For improving search engine recall rate and accurate rate, Using knowledge database method analyzed intelligent search engine technology and designed research ideas about intelligent search engine technology facing electronic commerce. Realized intelligent network ROBOT and realized individual search technology and E-mail push technology. The research provides information search service facing electronic commerce. Designing intelligent search engine raises recall rate and accurate rate of multiple search engine .

Keywords: Search Engine, Intelligent Search, Network ROBOT, E-MAIL Push , Electronic Commerce

1. INTRODUCTION

The fast development of Internet increases information quantity, Each people needs fastly find using information. The need of market stimulates to produce search engine technology and is perfected, and cultivated international great fame company of search engine technology on internationality, for example Yahoo, Altavisa and infoseek, and Google etc. Current search engine has following weaknes:

- (1) When search engine inquires for one topic, the result of query has very great number information and has so much rubbish information.¹
- (2) The display order of query result is in disorder, and has no classify.
- (3) It can't combine with electronic commerce. There are many success factors of electronic commerce, one of these is that customer can fast exactly find appropriate goods.

For improving search engine recall rate and accurate rate, Using knowledge database method analyzed intelligent search engine technology and designed intelligent search engine technology facing electronic commerce. The project has several initiatives: The first is the design idea of search engine which uses advanced information search methods and AI knowledge. The second is the realization and designing of intelligent search ROBOT facing electronic commerce. The third is the realization of individual character search engine and E-MAIL push technology. The research has real meaning in search engine field.

2. THE BASE CONSTRUCTION OF INTELLIGENT SEARCH ENGINE

(1) The base construction of search engine

Search engine mainly has four parts: Crawler ,Indexer,

Searcher and User Interface.[1]

Crawler is a kind of automatic network follow the tracks of program called for ROBOT. It is really an index file on network and software that it automatic follows the tracks the file of hypertext and cyclically searches all files. It crawls in network information space and visits common web sites and records network address. It indexes the content of web sites and constructs index file and forms index database to be indexed.

The function of indexer is to understand the information of crawler and takes out index item and builds up physical index database. Different network search tools have different index methods. Some software index all article. Some software index homepage address , title , specific paragraph , keywords, so database content has name of web sites , title , network address URL ,and the length of homepage and key words ,and hyperlinks and content introduction and abstract etc.

The function of searcher fast finds web sites according to user query and appraises relevance degree between web site and key words and orders the output result and realizes user feedback methods.

The roles of user interface inputs user query and displays the result of search and provides to user feedback method.

(2) The system structure of intelligent search engine

So far search engine has three types according to different tactics constructed[5]: The first search engine based on classify, for example Yahoo. Because the search engine used artificial classify method, the accurate rate was very high when search engine searched information. Its weakness was it used artificial classify method, it could not real time search to all information of network. It searched only in information of Yahoo searched, so the recall rate was not very well. The second search engine based on index text, for example Hotbot, its method was that ROBOT searched network information, and built index file to the information searched. This search engine efficiency was high than the first, but accurate rate was lower than the first. Because it used Robot to search information continually, the recall rate was high than the first. The third search engine was in view of concept, for example EXCITE, it searched not judging key words whether existed, otherwise using dictionary simply expand condition and using taking out of mode distinguished the relation of searching condition and web site. This is a kind of new technology, its accurate rate was lowest of three kinds of search engines, and recall rate was the highest. The traditional relation database could not deal with common knowledge to user, for example the "computer" and "electronic brain" etc. Though they are the same thing, but finding information to use key word "computer" might be lose the "electronic brain" information and recall rate was low. The reason lost information was the search engine lacked the ability to handle and understand knowledge. The key problem is to

* The project was aided financially by science and technology plan project of 2002 ShanDong province (2002-276-022090104)

advance the level from key words to intelligent knowledge and solves the computation method of relevant degree. In order to solve rubbish information and recall rate and accurate rate in multiple search engine, the project advanced a kind of new designing idea facing specific topic in search engine field.

The project combined classify of Yahoo and index method of Hotbot ROBOT with basing concept of EXCITE, built an efficient search engine and discussed it. The project advanced a new idea of search engine facing specific electronic commerce, and designed intelligent search engine ROBOT and individual index technology and E-MAIL PUSH technology. The intelligent search engine structure was as follows chart 1:

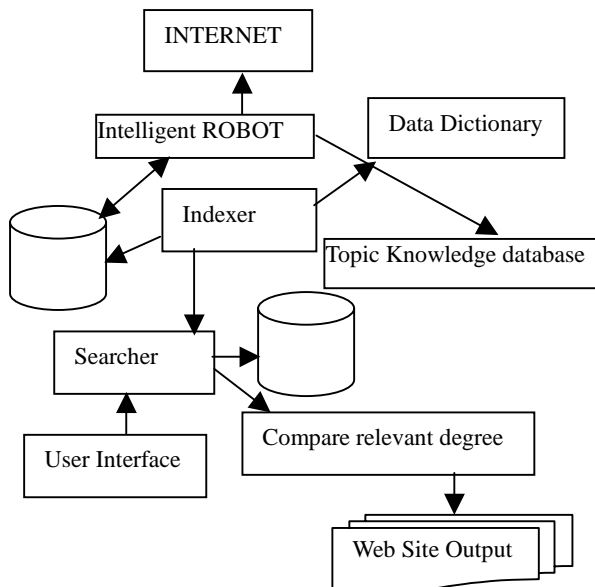


Chart 1 The Structure of Intelligent Search Engine

3. ANALYSIS AND REALIZATION OF INTELLIGENT SEARCH PRINCIPLE

(1) Analysis and realization of intelligent technology of network ROBOT[6]

The main function of network ROBOT automatically grabbed web sites and picked up information to describe web sites, and provided primitive data for search engine database. These data included HTML, TITLE, key words, main description and link numbers of HTML file[2]. It obtained web sites through its sent request to other computers, and searched cyclically all files directed by hyperlinks.

ROBOT was a software system, it included four parts: visiting control part, path selecting part, file visiting part, database handling part. The relation among parts was as follows chart 2: ROBOT started from one group URLs defined. The group URLs might be one group web sites that they were often visited or ROBOT started from the URL defined by user. When ROBOT visited one web site, it recorded all new URLs in the web site and visited continually with new URL until they had not new URL or arrived at a limit. ROBOT not only took out hyperlinks but also filtered files that were not appropriate. Every hyperlinks were saved according to the analysis of topic knowledge database, and built index for WEB search engine and produced local database. Some ROBOTs built index according to HTML title, some ROBOTs

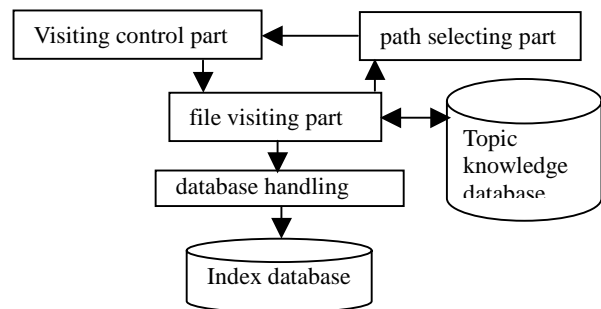


Chart 2 WEB ROBOT principle facing topic

analyzed all web sites and built index on words. The project built index according to all words.

The ROBOT grabbed specific information defined by user facing electronic commerce field, and built index on words and used by user when requested.

The project took out hyperlinks using JAVA language. The input and output was abstracted as stream by JAVA language. It obtained head field and text with `getContent` and `getHeaderField`, and `getInputStream` and `getOutputStream` methods. Because TCP protocol transfers byte as stream, it can correct display only in character type.

Using `InputStreamReader` class will change present input stream into character stream. In order to effectively read byte, using `Java.io.BufferedReader` class read text from input stream and put text into buffers.

Created input stream was as follows:

```
InputStreamReader in = new BufferedReader(in);
BufferedReader display = new BufferedReader(in);
Created URL connecting sentence was following:
URLConnection urlConnection = url.openConnection();
```

(2) ROBOTS search tactics

Each ROBOT defines visiting order otherwise repeat labor will produce. ROBOT search tactics is how transfer to next file after ROBOT searches a file. It mainly has following several tactics:

IP address tactics: It is simplest tactics. The method is to give ROBOT a initial address, then ROBOTS search every files according to IP address adding. ROBOTS don't consider other links directed. The merit is wide searching. The weakness is not suited to bigger wide searching.

Deep preference search tactics: It starts from initial web site until files don't include hyperlinks, then returns to the layer another web site to continue search other hyperlinks. It ends until not having other hyperlinks. The merit easily find new web site, but the growth of information wide is relative slow.

Wide preference search tactics: It finishes searching all hyperlinks in a web site, then continues next layer until finding the lowest layer. It can well solve search wide problem. The weakness is that ROBOTS take a long time to arrive deeper web sites.

Here combined deep search tactics with wide search tactics facing topic ROBOT. It used both merit to remedy opposite weakness. It is the direction of search engines. Here constructed search engine wandered on network and counted

web site hyperlinks and searched web site. Robot repeated as above steps. Robot search tactics algorithm was as follows[5]:

```

Begin
  Let I be a list of initial URLs;
  Let F be a queue;
  For each URL in I
    Enqueue(I,f);
  End
  While !Empty(F)
    u ← Dequeue(F);
    d ← Get(u); //request document d pointed
    // by u with special title
    store d;
    Extract the hyperlink from d;
    Let U the set of URLs cited in these hyperlinks;
    For each URL u in U
      Enqueue(u,F);
    end
  end
end

```

Enqueue option added a new URL in arrange. Dequeue option popped up the first URL from the arrange. Dequeue option did not really delete the first URL, only tagged it. After all URLs were tagged, the condition Empty(F) was true.

(3) Web site importance analysis

Efficiency of search engines depends on index quality. So far there are two computation method of relevant degree:

Web site links structure analysis method

Using each other links among web sites describes a web site 'quality'(PageRank).If a web site is cited by more other web sites, its PageRank(Importance) is more high.

PageRank description is as follows[3]: To suppose there are n web sites T_1, T_2, \dots, T_n , they directs web site W cited. Parameter d is attenuation factor. d value is between 0 and 1. According to experience, d value is 0.85. $C(W)$ is link number of directing to outside web site. PageRank of W defined as follows:

$$PR(W) = (1-d) + d(PR(T_1)/C(T_1) + \dots + PR(T_n)/C(T_n)) \quad (Eq1)$$

PageRank forms a probability dispersed. Sum of all web sites PageRank is 1. Each pagerank value is a standard cited.

Using word frequency method of statistics analysis

According to one word appearance frequency and position and index word weighted, relevant degree of the web site is computed.

Here used word frequency to compute relevant degree of web site. The realizing method was as follows. The location field of WordSearch table saved index information. The appearanceNumber field saved web site number searched. The location field format was as follows: ID, Importance ID, Importance ID, Importance ... In Location field the importance computation was as follows:

$$\begin{aligned} \text{Importance} = & \text{TitleImportance} * \text{TitleCount} + \\ & \text{BodyImportance} * \text{BodyCount} + \\ & \text{DescriptionImportance} * \text{DescriptionCount} + \\ & \text{KeyWordImportance} * \text{KeyWordCount} \end{aligned} \quad (Eq2)$$

The variable with Importance was weighted. The variable with Count was word frequency.

The insufficiency both relevant degree computation was that the relevant degree was determined before user used it. It was the result according to predetermine algorithm computing. Though the factor of key words number appeared and position

and hyperlinks might be reflected the relevant degree between key words and web sites, but it was not overall. It did not include user factor selected. After user searching, user can really know the relevant degree between web sites and key words. So people factor must be considered in relevant degree computation. Using people intelligence finishes that computer can not finish intelligence. Our method was as follows.

(4) Adding artificial intelligence to construct network individual service model

Constructing mathematics model[4]

After users browsed the query results that indexer returned, they selected to browse the query results. Because indexer each time returned thousands of query results, the results were very poor. They decided content wanted according to the snapshot of results. So we consider query results are relevance to key words if users select one result. It is equal to give a supporting ticket. So the relevant degree between key words and the web site is raised. If users do not select the select, it is considered not relevance between query result and key word. It is equal to give a opposite ticket. So the relevant degree between key word and the web site is reduced.

Supposing function $P(X)$, in $P(X)$, X is relevant degree computed between one key word and one web site. $P(X)$ is selecting probability of all users in the relevant degree. The probability is a average probability of all users selected. Using average probability may avoid to effecting personal option to final result. We supposed the accurate solution was X^* between the key word and the web site. The problem is turned to compute X^* mathematics problem.

First analysing X^* has how character. As X^* is accurate value of relevant degree between the key word and the web site, $P(X)$ is toward to steady near X^* . It is that order of query results has a little influence to user selecting. We know function derivation reflects function value changing with X changing, so evaluation of X is the X derivative near zero. As we do not know $P(X)$ expression, so we can not compute X^* through derivation. But the derivation may use difference quotient to replace it. Difference quotient is expressed by $f[X_1, X_2]$.

$$f[X_1, X_2] = (P(X_2) - P(X_1)) / (X_2 - X_1) \quad (Eq3)$$

Supposing there is a value k , when different quotient is less than k , we think $P(X)$ is towards steady. As result X is the lowest X point difference quotient that $P(X)$ is lower than k . X value segment is as follows chart 3.

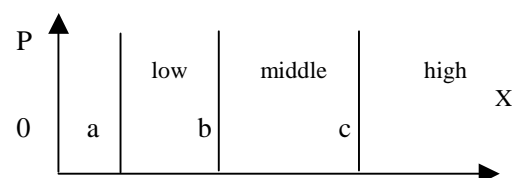


Chart 3 X Value Segment

As above built mathematics model considering user selecting probability. It computed the lowest relevant degree X that made difference quotient steady. The X was the best relevant degree X^* wanted.

In individual character dealing model, indexer function acquired user information and first searched index database and display front ten results for users to select it. The front ten results can be deal by dividing words. To count the repeat words and to select front order words, new words are

transferred to indexer. URL sites were renewed ordered as the relevant degree between key word and web site. It could make user query more efficiency. This was equal to add user selecting factor in relevant degree computation. Using artificial intelligence finished information searching.

Individual character PUSH service model

PUSH technology is a kind of new information issued technology. It advanced a new service model as a new technology on Internet. The service model has activity. The service model may push interested information to user. Push technology has two models: One is frequency channel PUSH technology. This technology defines some web sites as frequency channel in browser. Users may see interested information that network sends as if they selected movies. The another is mail PUSH technology. The technology is active to issue information to user in list table with e-mail mode.

Our realization took E-mail PUSH technology mode to push selected information to user e-mail directed. The flow was as follows:

- 1) Fill in user subscribing to bill. The subscribing to bill included user personal files and interested information type and push time etc. Then it was submitted to provider.
- 2) As the same time user searched topic information through index interface, user may select whether the information was pushed. If users selected PUSH, it might realize artificial PUSH or automatic machine PUSH until information was pushed to user e-mail.

Here realized intelligent artificial PUSH technology. It really combines intelligent search with electronic commerce, and realized information feedback on network between buyer and seller and produced more market effect.

4. CONCLUSIONS

As above is research model that we advanced intelligent search technology facing electronic commerce. The intelligent ROBOT can really grabbed information to index database facing topic. The technology that the model advanced had been realized. Through we tested the intelligent search field to 'computer' and 'software' and so on electronic commerce field, the fact was that we added artificial service model in three thousands results of query returned, the result returned two hundreds of used information. The accurate rate was well raised. We may select used information PUSH to user e-mail. It is really combined intelligent search with electronic commerce. It proves the program is feasible. The system needs more perfect in real application. We hope our research can give much contribution for Internet and search engines technology.

5. REFERENCES

- [1] Lawrence, Giles C L. Context and Page Analysis for Improved Web Search. [J] Internet Computing, 1998(7/8):38
- [2] Koster M. The Web Robots FAQ... [OL]. <http://info.webcrawler.com/mak/projects/robots/faq.html>. 1998-5-14
- [3] Page L, Brin S, Motwani R, Winograd T. The PageRank Citation Ranking: Bring Order to the Web. January 1998
- and July 2001 at <http://www-db.stanford.edu/~backrub/pageranksub.ps>
- [4] Li xiao-ming, Advances of Search Engine and WEB Mining Higher education press [M] March of 2003: pp34--40, in Chinese
- [5] Bi qiang Develop and utilizing of network information resources Scientific press [M] October of 2002: pp128-129, in Chinese
- [6] TONG Xiao-jun, Research and design of Intelligent ROBOT facing topic, Journal of electronics & information technology. [J] 2003 VOL.25 Suppl. Pp584-590, in Chinese

Tong Xiaojun is a vice professor, head of information secure teaching and research section of department of computer, HarBin University Of Technology (Weihai). She graduated from Computer department of North East university in 1988, was a visiting scholar of DELFT Technology of Netherland (1996-1997), worked at I-SYSTEM software company in JAPAN (1999-2000). Her research direction is computer network, information security, electronic commerce and database. The project was aided financially by shandong province science and technology plan project of 2002 (2002-276-022090104).



Communicate address: Weihai city Gao Qu HarBin Institute Of Technology (Weihai) in school 20#406 room
Postcode: 264209

Metadata Catalog Service for Geographic Information Resource

Xu Kun, Liao Husheng, Du Jinlian

College of Computer Science and Technology

Beijing University of Technology, Beijing 100022, P. R. China

Email: xu.kun@emails.bjut.edu.cn liaohs@bjut.edu.cn dujinlian@bjut.edu.cn

ABSTRACT

This paper presents a framework of Geo-Information Catalog Service System. Catalog Service System uses for the administration of the metadata in georesources exist in Internet to implement the share of georesources. This paper designs the framework of Catalog Service System based on the research about metadata and Metadata standard, experience of Metadata Systems, and application requirement. Further, this paper design the strategies of metadata based on XML in description, transform, query and store. Finally, this paper provides an implementation of the Catalog Service System named CAT System, which is used to build a GIS Decision-Making System with WFS and WMS together. The GIS Decision-Making System gives GIS technology supports to Beijing E-Gov.

Keywords: Catalog Service , Goeresource , Metadata , XML

1. INTRODUCTION

Along with quick development of technology in network and database, the GIS(Geography Information System) also experienced the server/terminal system, desktop workstation system, LAN distributed system, to web-accessible geo-processing services. Since geographic information resources lie in various sorts of database and stored in different formats, belonging to different individuals and group, thus how to obtain these abundant geographic information in the Internet is a problem that needs to be solved urgently. This paper introduces a metadata catalog service framework for geographic resources, to provide with a mechanism to centralize and standardize information about GIS data. With the catalog service, data user can search and quickly find information about data. A data steward who is responsible for maintaining data can publish information about their data. Application developer can automate data access and management by querying the metadata catalogs. The main features of the framework are as follows:

- 1) Providing with online catalogs for geographic information and web-accessible geo-processing service, such as various GIS layer, Web Feature service and Web Map service.
- 2) Customization to support various standards of metadata.
- 3) Conforming to the OpenGIS Catalog Service Implementation Specification, supporting the OGC_Common language and spatial operation.
- 4) Supporting distributed search for metadata information on geo-data and geo-processing service.

We have implement a metadata catalog server for geographic resources that bases on the Web Service. The software realizes resource management with various metadata standard and resource discovery with the distributed search..

OpenGIS catalog service

The OpenGIS Specification represents the collective wisdom of the OGC(Open GeoData Consortium) Technical Committee. Developers building systems with OpenGIS Specification conformant interfaces will create middleware, component-ware, and applications that can handle a full range of geo-data types and geo-processing functions. Users of these systems will be able to share a potentially huge networked data space, even though the data may have been produced at different times by unrelated groups using different production systems for different purposes and may in fact still reside under the primary control of the systems used in their production. Legacy geo-data held in systems with OpenGIS Specification conformant interfaces will be accessible by other software with OpenGIS Specification conformant interfaces.

The OpenGIS Specification defines a union 3-tier structure model for all types of services. These servers provide service by exposing their Web service interface. Clients can communicate with server through XML or GML. The OpenGIS Catalog Service Specification version 1.1.1 documents industry consensus regarding an open, standard interface to online catalogs for geographic information and web-accessible geo-processing services. Industry agreement on a common interface for publishing metadata and supporting discovery of geo-spatial data and services is an important step toward giving Web users and applications access to all types of "where" information.

2. DESIGN OF CATALOG SERVICE

To satisfying the needs of opening and interoperability, our catalog service pick up an interface outward to provide by Web service, the client software as long as match to this interface can adjust to use the server software. Supporting XML is not only in Web service but also in describing and querying metadata. Normally the structure of metadata is a hierarchy, which is same with XML. A lot of metadata standards support XML, and the power will be strengthened in the future. These standards provide XML schemas to describe the metadata's structure. Search result be organized with the XML document format, a client application can decompose search result according to the XML Schema.

Since different geographic resources from different organizations may use different metadata standards. It is necessary to expand a new metadata standard into a catalog server. The customization for metadata standard is defined in a XML schema, and a XML schema editor is provided for importing metadata standard.

A query interface is used to locate geo-data and geo-processing service. The search should contain several search criteria to meet user's need, including both spatial criteria and textual criteria. For the description of those spatial criteria, our catalog service adopts geometry type and spatial

operation based on OpenGIS Specification.

Owing to the network existing huge geographic resources in quantity, the distribute type search of the geographic resources can be made to find a better way to organize and manage metadata, reduce the cost of query and promote system stability. Distributed query generally contains two kinds of circumstances: One is that all query requests should be sent to a certain server named gateway and the gateway transfer this request to servers which are registered in the gateway, another is that a client can send a query request to any server reachable and this server transfer request to sub-servers directly. Each two kinds of methods has its merits and shortcomings, we adopt the second method.

3. SYSTEM FRAMEWORK

As shown in following diagram, the Metadata Catalog Service system adopts 3-tier construction. Users can access this system directly through a client application, or users can use browser to access to the system by a Web server. The composing of the server is shown in the center dotted box. The Catalog Service Interface is separated with logic business modules, which will make it easy to update and maintain system and implement multi-interfaces.

The Session Manager module is a processing center to respond requests from different sessions and to keep status data of sessions and requests. According to requests, this module will invoke relevant function modules: Index Manager Module, Manage Module and Query Module.

According to the configurations set by data steward, the Index Manager Module will select a part of metadata items and create an index table in a RDBMS. A metadata instance is a XML document, In common, the hierarchy of XML Schema document regarded as a metadata standard is very complex. If

the XML Schema document is transformed to relational tables, the result is a collection of relational tables. Relations among these tables are irksome and most of columns in tables will never be used in search criteria. And these may lead to storage expanse and inefficient query. So data steward is responded to select a part of XML element and attribute for building relational tables named index table. With index tables be established, the relations between column of tables and element or attribute are build. Thus search may be performed through not only OGC_Common Language but also standard SQL.

Main functions of the Manage Module are to add, modify and delete metadata of geographic information resources exist in network. While adding a metadata instance, these data relate to columns of index tables will be picked up from XML document and insert into index tables.

The Manage Module is the core for whole system. While receiving a query request from a client, this module will parse the query condition statement in OGC_Common Language, then seek out the parts contain geometry type and spatial operation, and transform the left to SQL statement, get query result from RDBMS through SQL statement, and finally perform the spatial operation to filter query result.

For distribute search, the server should send query request to any sub-server that have been registered, which is another metadata catalog server in the Internet, and organize these results from sub-server. How to check in a sub-server is using a server configuration application that we have implemented. It is necessary to register sub-servers need to access directly for a server. If every server's configuration has done, there will be a graphic about relations among these servers. A global exclusive request identify is used for each catalog server for avoiding reduplicate query and a query loop within these servers.

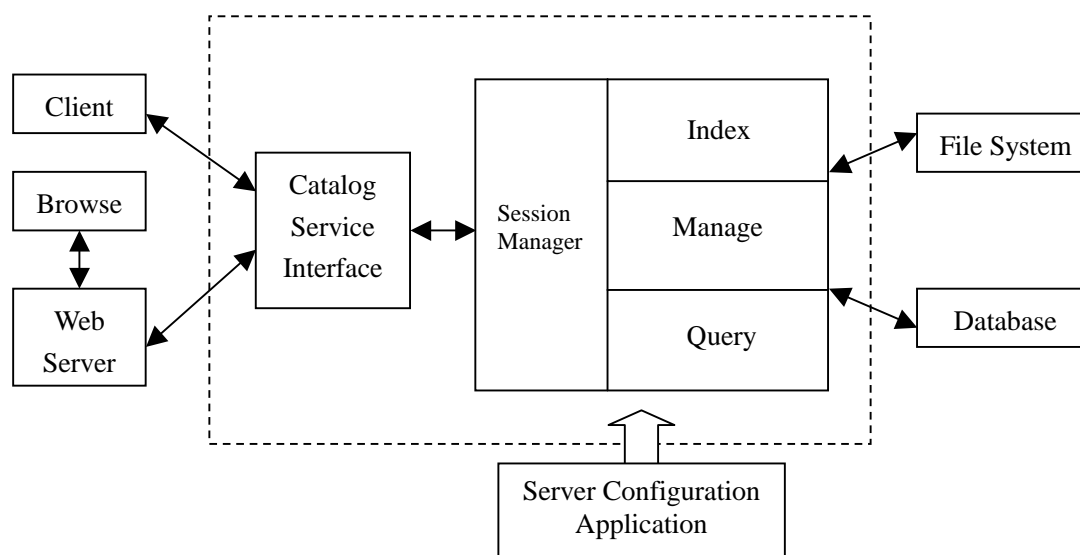


Fig.1 Catalog Service Framework

4. OGC_COMMON LANGUAGE

The OpenGIS Specification defines a special query language named OGC_Common Language for Catalog Service. It is the biggest characteristics point that geometry type and spatial operation can be used to create a query condition expression.

The OGC_Common Language defines 8 basic types as follows: the integral, real, boolean, string, date, time, null, timestamp, and still defining geometry types such as Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, GeometryCollection, Envelope. Using the Representation of OpenGIS Specification to represent the geometry types. According to norm of several why object

model, the Point, LineString, Polygon, Envelope and GeometryCollection respectively five kinds of geometry object, but the MultiPoint, MultiLineString and MultiPolygon means the collection of Point, LineString and Polygon. The spatial operations only include related spatial operations, these operations are two kinds: One is the operation as follows between two geometry objects: EQUAL, DISJOINT, INTERSECT, TOUCH, CROSS, WITHIN, CONTAINS, OVERLAP, etc. And WITHIN, BEYOND operations are the operations between a geometry object and number.

The OGC_Common Language also included and, or, not, brackets, comparison, the faintness matches, the empty value judges to carries with the space definition that calculate the type of expression. Customer can expand the syntaxes base on OGC_Common Language according to actual require. To do so can enlarge the functions without changing the interface.

5. IMPLEMENTATION

We have implemented a prototype of the Metadata Catalog Server, on Windows .NET platform in C# programming language. We have realized the local and distributed search using OGC_Common Language, customization by metadata standards and the spatial operation on Envelope type.

6. SUMMARY

The design points of Metadata Catalog Service consist in regarding the OpenGIS specification as foundation, enlarge to XML support, support the metadata standards of geographic information resources defined by customers, realize the distributed search. All of these are in order to guarantee customers can accurately find the resources the customers want to visit in time. At the same time, the system also provided certainly flexible, in order to satisfy the different application need.

7. REFERENCES

- [1] OpenGIS Consortium, OpenGIS Catalog Interface Implementation Specification, Wayland, Massachusetts. 1999, <http://www.opengis.org/techno/specs.htm>.
- [2] OpenGIS Consortium, OpenGIS Service Architecture, Wayland, Massachusetts, 2002, <http://www.opengis.org/techno/specs.htm>.
- [3] OpenGIS Consortium, The Catalog Services, Wayland, Massachusetts, 2002, <http://www.opengis.org/techno/specs.htm>.
- [4] ESRI, Metadata and GIS – a ESRI write paper, 2002, <http://www.esri.com/library/whitepaper/pdfs/metadata-and-gis.pdf>.

Xu Kun received his B.S. degree and M.S. from Computer Institute of Beijing University of Technology in 2001 and 2004. His research interests include spacial database and metadata.

Liao Husheng graduated from Beijing University of Technology in 1977, and received the M.S. degree from Tsinghua University in 1981. He is a Professor in Computer

Institute of Beijing University of Technology. His research interests include partial evaluation, data integration, Java and spacial database.

Research of Comparing CORBA with DCOM *

Wang Jingyang¹, Wang Xiaohong¹, Yuan Dun², Wang Jianxia¹, Ma Xiaojuan¹

¹ Hebei University of Science and Technology

Shijiazhuang Hebei 050054, China

² 8357 Research Institute of China Aerospace Science & Industry Corp

No.69, Huangwei Road, Hebei District, Tianjin, China

Email: jingyangw@hebust.edu.cn Tel: 0311-8613336

ABSTRACT

This article discusses the principle and architecture of the two famous distributed object technologies CORBA and DCOM, provides their common merits and shortcomings. It also compared their aspects of the ability of astride-platform, integrating different language, invoking method and communication protocol. Thus, the distributed object technologies CORBA and DCOM can be fully understood and their applications can be grasped through comparative study of their similarities and differences.

Keywords: CORBA, DCOM, ORB, Stub, Skeleton, Distributed programming.

1. INTRODUCTION

Distributed object computing extends an object-oriented programming system by allowing objects to be distributed across a heterogeneous network, so that each of these distributed object components interoperate as a unified whole. These objects may be distributed on different computers throughout a network, living within their own address space outside of an application, and yet appear as though they were local to an application. Two of the two most important and widely used distributed object systems are Object Management Group (OMG)'s the Common Object Request Broker Architecture (CORBA) and Microsoft's Distributed Component Object Model (DCOM). Both are being used in the industry for various applications ranging from e-commerce to health care. Selecting which of these two distribution mechanisms to use for a project is a tough task. This article gives a detailed comparison of CORBA and DCOM ^[1].

2. PRINCIPLE AND ARCHITECTURE

CORBA is an industry standard developed by the OMG to aid in distributed objects programming. It bases on object-oriented and its basic mechanism is client/server model. CORBA provides a service platform, through which heterogeneous networks can access each other and work harmoniously in distributed environment. CORBA relies on a protocol called the Internet Inter-ORB Protocol (IIOP) for remote objects. Everything in the CORBA architecture depends on an Object Request Broker (ORB). The ORB acts as a central Object Bus over which each CORBA object interacts transparently with other CORBA objects located either locally or remotely. Each CORBA server object has an interface and exposes a set of methods. To request a service, a CORBA client acquires an object reference to a CORBA server object. The client can now

make method calls on the object reference as if the CORBA server object resided in the client's address space. The ORB is responsible for finding a CORBA object's implementation, preparing it to receive requests, communicate requests to it and carry the reply back to the client. A CORBA object interacts with the ORB either through the ORB interface or through an Object Adapter - either a Basic Object Adapter (BOA) or a Portable Object Adapter (POA). Since CORBA is just a specification, it can be used on diverse operating system platforms from mainframes to UNIX boxes to Windows machines to handheld devices as long as there is an ORB implementation for that platform ^[2]. The CORBA ORB architecture is shown in Figure 1.

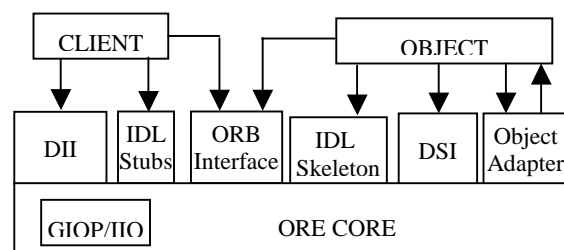


Figure 1 CORBA ORA architecture

DCOM extends the Component Object Model (COM) to support communication among objects on different computers—on a LAN, a WAN, or even the Internet. With DCOM, your application can be distributed at locations that make the most sense to your customer and to the application. DCOM is often called 'COM on the wire', supports invoking remote objects by running on a protocol called the Object Remote Procedure Call (ORPC). This ORPC layer is built on top of DCE's RPC and interacts with COM's run-time services. A DCOM server is a body of code that is capable of serving up objects of a particular type at runtime. Each DCOM server object can support multiple interfaces each representing a different behavior of the object. A DCOM client calls into the exposed methods of a DCOM server by acquiring a pointer to one of the server object's interfaces. The client object then starts calling the server object's exposed methods through the acquired interface pointer as if the server object resided in the client's address space. As specified by COM, a server object's memory layout conforms to the C++ vtable layout. Since the COM specification is at the binary level it allows DCOM server components to be written in diverse programming languages like C++, Java, Object Pascal (Delphi), Visual Basic and even COBOL. As long as a platform supports COM services, DCOM can be used on that platform. DCOM is now heavily used on the Windows platforms ^[3]. The architecture of DCOM is shown in Figure 2.

* This paper is supported by the fund of Hebei University of Science and Technology (Grant NO. xl2003132)

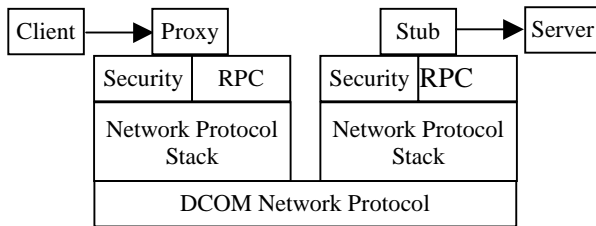


Figure 2 DCOM architecture

3. CORBA VS DCOM

3.1 Common Merits and Shortcomings

Both CORBA and DCOM extend and develop the theory of Remote Procedure Call (RPC), use local languages for encapsulation with alternation among distributed applications and make client object invoke the methods in remote server as if they were local to an application, which it has good hyalinization and performance. To programmers, there no difference between creating distributed applications and local applications, so we needn't study and use complex network API. So we can just use the thinking of programming local applications to develop distributed applications, and we can take most energy to design the application's interfaces and arithmetic. Also both of them adopt an object-oriented technology. In software analysis, designing, programming, maintenance and so on, the object-oriented system has much more advantages than traditional structural system. The primary unit of remote interfaces between distributed programs is objects, and remote request is invoking the methods of objects. Both of them support many object-oriented technologies such as encapsulation, multimode and so on. Otherwise CORBA support inheritance, but DCOM do not support inheritance. Therefore severely speaking DCOM does not completely adopt all object-oriented characteristics.

Although both of them adopt object-oriented technology and remote invoke objects as in local space, they only have a remote server's reference. Transmitting remote server's object to client is quite important function of distributed software, but DCOM and the version of CORBA before 3.0 don't have the function. It is a question waiting for us to solve. CORBA3.0 has the ability of transmitting objects through value, but can't completely solve the problem and is just a little better than old versions.

3.2 Spanning Platforms and Diverse Languages

CORBA's application is independent which is not related to languages, developers and operating systems. Because CORBA defined a set of interfaces not related to languages and environment, any application, software system or tool can easily integrate CORBA system if they are suitable for the criterion of interfaces. There are CORBA products in any main operating systems and languages. CORBA could access objects written by diverse programming languages like C++, COBOL, Small talk and Java. According to the protocol of IIOP, CORBA ORB can get and operate remote objects. Also, client application can be developed using diverse languages and the language of developing client may be different from server language.

DCOM is Microsoft's distributed component model, and it is mainly running on the Microsoft operating system. Notwithstanding Microsoft is cooperating with another

developer, DCOM will be moved on other operating systems. Because DCOM is defined and controlled by a single developer, users' choices area will be confined. Meanwhile, DCOM is lack of many supporting platforms. Therefore, the reuse of codes and the extensibility of DCOM could be constricted. Like CORBA, DCOM module can be developed with diverse programming languages. However Client and DCOM objects must adopt a uniform idea to depict interface (Standard binary system format). Depending on the interface definition, clients can invoke server methods directly not mention the program languages of DCOM object.

3.3 Invoking Method

Both DCOM and CORBA support static and dynamic invocation of objects. It is a bit different than how CORBA does through its Dynamic Invocation Interface (DII) or Java does with Reflection. To DCOM, for the static invocation to work, The Microsoft IDL (MIDL) compiler creates the proxy and stub code when run on the IDL file. These are registered in the systems registry to allow greater flexibility of their use. This is the vtable method of invoking objects. For dynamic invocation to work, DCOM objects implement an interface called IDispatch. As with CORBA, to allow for dynamic invocation, there has to be some way to describe the object methods and their parameters. In DCOM, an object whose methods are dynamically invoked must be written to support IDispatch. This is unlike CORBA where any object can be invoked with DII as long as the object information is in the Implementation Repository.

CORBA support multiple-inheritance at the IDL or interface level. One difference between CORBA IDLs and DCOM IDLs is that CORBA can specify exceptions in the IDLs while DCOM does not. In CORBA, the IDL compiler generates type information for each method in an interface and stores it in the Interface Repository (IR). A client can thus query the IR to get run-time information about a particular interface and then use that information to create and invoke a method on the remote CORBA server object dynamically through the Dynamic Invocation Interface (DII). Similarly, on the server side, the Dynamic Skeleton Interface (DSI) allows a client to invoke an operation of a remote CORBA Server object that has no compile time knowledge of the type of object it is implementing^[2].

3.4 Difficulty of Development

CORBA possess some characteristics of spanning platforms and languages. It means that the process of development is quite difficult, which functions can support all kinds of traditional languages and platforms. If developing applications with CORBA, we must grasp program language, IDL language, ORB, BOA, interfaces and so on. DCOM is mainly used in the communication between Windows programs. One of DCOM's advantages is that many tools could set up COM and DCOM components, including C++ tools (such as Visual C++), RAID (such as Visual Basic, Delphi and Power Builder). Otherwise, a lot of ActiveX components, which are already set up and commercial could be used.

3.5 Client and Server's Detached Extent

CORBA adopts Broker concept. Broker has some effects: completing services mapping by client, automatically sending request, searching server objects and take back the result from the server. Client programs no longer directly contact with servers, just alternating with ORB, which client and server are

completely separated. Client programs of DCOM can transfer DCOM objects through pointer to interface. You must instance DCOM and then can use the interface. So client and server could communicate directly^[4].

3.6 Communication Protocol

DCOM Uses the Object Remote Procedure Call (ORPC) as its underlying remote protocol. However CORBA Uses the Internet Inter-ORB Protocol (IIOP) as its underlying remote protocol.

3.7 Affording Services

CORBA defines many object services, including naming service, event service, life cycle service and so on. These services are necessary functions during developing application. The services of CORBA can supply great convenience for distributed software developers and make them concentrate energy on developing functions. The management of DCOM's life cycle is a quotation-counted mechanism, which is realized by AddRef and Release methods of IUnknown's interface. Otherwise, DCOM takes use of "point to point" mechanism for supplying transmissive path and Microsoft Transaction Service (MTS) can supply the function of transaction service and secure service for DCOM.

3.8 Security

Comparing with DCOM, CORBA could supply a more perfect secure model. The secure services of CORBA could supply identification, delegation, encrypt, secure area and verification services with following network secure rows. DCOM adopts Windows NT secure system. If to not Window platforms, DCOM would use the secure system of these platforms and supply a secure measure that it's compatible with Windows NT.

3.9 Multithreading Support

CORBA can support multithreading, which it allows that many client programs invoke some CORBA server objects at the same time and server create a multithreading for every client to deal with its requests. But comprehensive EMS memory variable and data should be protected in programs. However DCOM support Apartment Model of multithreading mode, which process can have many threads but the object of DCOM can't have many threads. So every DCOM object can only run in one thread.

3.10 Cooperation of CORBA and DCOM

Because DCOM of Microsoft has a lot of users and it is also an important part of distributed object modes. OMG issued COM/CORBA cooperated protocol standard and COM/CORBA is already written in CORBA2.0. Then official standard will be approved. So it is very import that COM and CORBA cooperate with each other in main distributed platforms.

4 CONCLUSIONS

The article is comparing two current famous distributed objects CORBA and DCOM. They have different characteristics. However, in general, CORBA has more advantages than DCOM. In spanning platform systems, CORBA is better. But in Microsoft technological systems, DCOM is the best choice. The two distributed developing technologies—DCOM and CORBA will exist for a long time and they will have more cooperation. With the development of distributed technologies, the difference between them will reduce constantly.

5 REFERENCES

- [1] Jason Pritchard. Essentiality and cooperation of CORBA and COM [M]. Tsinghua University Press, June 2002 (in Chinese)
- [2] OMG. How do remote invocations work [EB/OL]. <http://www.omg.org/gettingstarted/corbafaq.htm>
- [3] Microsoft Corporation. DCOM Architecture [EB/OL]. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomarch.asp
- [4] OMG. What is CORBA good for [EB/OL]. <http://www.omg.org/gettingstarted/corbafaq.htm>



Wang Jingyang (1971-), male, he is a lecturer of Hebei University of Science and Technology. He graduated from Lanzhou University with specialty of Computer Software Science in 1995. He has published two books, over 10 Journal papers. His research interests are in Network and Database application technology.

The Summarize of JAVA Platform for Web-based Computing

Hui-Ying Xu ^(a), Xin-Zhong Zhu ^(b)

College of Computer Science and Engineering, Zhejiang Normal University

Institute of Computer Science Studies, Zhejiang Normal University

Jinhua, Zhejiang 321004, China

Email: jhxhy@mail.zjnu.net.cn Tel.: (86) 05792282145

ABSTRACT

In recent years, the volume of datasets in modern large-scale scientific researches, information services and digital media applications is growing explosively, and the research about data grid technology becomes the new hotspot in the computer science all over the world. Since 1990 some great progresses, about the basic theoretical research and test-bed environment construction, have been made all over the world. And now researches on the Grid computing and web-based simulation and computing technologies have been made and have already made a rapid progress.

With the development of these technologies, the Java platform has been expanded rapidly and used widely. Now the Java technology and platform have applied widely in grid computing and services, distributed computing, Web services and Web-based simulation and computing, etc.

Keywords: Java, JVM, EJB, J2EE, J2SE, J2ME

1. INTRODUCTION

Java technology is computer software that helps you get connected and makes being connected more exciting. Invented by Sun Microsystems in 1995, Java technology lets devices of all kinds run just about any kind of program, giving you the cool games, tools, and information you want most. With the development of the Java technology and the widely usages on it, Java technology is everywhere. It's embedded in 150 million mobile phones; it's in PDAs and pagers; it's inside video games, TVs, and Web sites. It's pre-installed on personal computers. It's even in cars and on the planet Mars. Meanwhile, the Java brand, with its well-known cup and steam, is also everywhere. It's one of the most widely recognized technology brands in the world! Look for it on games, handsets, and Web sites that are powered by Java technology. Wherever you find the cup and steam, you'll find Java technology and a great digital experience.

Java technology is both a programming language and a platform. And now Java platform has been well-known platform and used widely in many fields all over the world.

In Sect. 2, we introduce some Java platforms. In Sect. 3, the applications of Java platform in distributed computing, mobile computing and Web-based simulation and computing are presented. And we conclude in Sect.4.

2. OVERVIEW OF JAVA PLATFORM

The Java Platform is a new software platform for delivering and running highly interactive, dynamic, and secure applets and applications on networked computer systems. But what sets the Java Platform apart is that it sits on top of these other platforms, and executes byte codes, which are not specific to any physical machine, but are machine instructions for a virtual machine. A program written in the Java Language compiles to a byte code file that can run wherever the Java Platform is present, on any underlying operating system. In other words, the same exact file can run on any operating system that is running the Java Platform. This portability is possible because at the core of the Java Platform is the Java Virtual Machine.

While each underlying platform has its own implementation of the Java Virtual Machine, there is only one virtual machine specification. Because of this, the Java Platform can provide a standard, uniform programming interface to applets and applications on any hardware. The Java Platform is therefore ideal for the distributed application, where one program should be capable of running on any computer. The Java Platform is designed to provide this "Write Once, Run Anywhere" capability.

Developers use the Java Language to write object-oriented, multithreaded, dynamically linked. They compile once to the Java Platform, rather than to the underlying system. Java Language source code compiles to an intermediate, portable form of byte codes that will run anywhere the Java Platform is present.

The Java Platform create environment for writing distributed applications. The Java Platform enables to write distributed application not only on traditional computing device. The application can run from embedded device, like Smart Card to high server environment.

Recognizing that "one size doesn't fit all," Sun has grouped its innovative Java technologies into three editions: Java 2 Platform, Micro Edition (J2METM technology), Java 2 Platform, Standard Edition (J2SETM technology), and the Java 2 Platform, Enterprise Edition (J2EETM technology). Each edition is a developer treasure chest of tools and supplies that can be used with a particular product.

In next sections, we will introduce these Java platforms respectively.

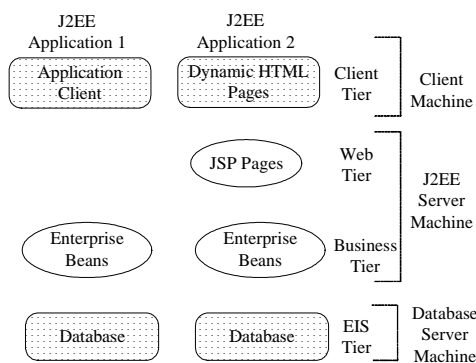
2.1 Java 2 Platform, Enterprise Edition (J2EE) Overview

J2EE technology simplifies enterprise applications by basing them on standardized, modular and re-usable components Enterprise JavaBeansTM (EJBTM), providing a complete set of

services to those components, and handling many details of application behavior automatically. By automating many of the time-consuming and difficult tasks of application development, J2EE technology allows enterprise developers to focus on adding value. That is, enhancing business logic, rather than building infrastructure.

And the J2EE platform uses a multitiered distributed application model for enterprise applications. Application logic is divided into components according to function, and the various application components that make up a J2EE application are installed on different machines depending on the tier in the multitiered J2EE environment to which the application component belongs. Figure 1-1 shows two multitiered J2EE applications divided into the tiers described in the following list:

- Client-tier components run on the client machine.
- Web-tier components run on the J2EE server.
- Business-tier components run on the J2EE server.
- Enterprise information system (EIS)-tier software runs on the EIS server.



A J2EE application can consist of the three or four tiers shown in Figure 1-1. But J2EE multitiered applications are generally considered to be three-tiered applications because they are distributed over three different locations: client machines, the J2EE server machine, and the database or legacy machines at the back end. Three-tiered applications that run in this way extend the standard two-tiered client and server model by placing a multithreaded application server between the client application and back-end storage.

The primary technologies in the J2EE platform are: Java API for XML-Based RPC (JAX-RPC), JavaServer Pages, Java Servlets, Enterprise JavaBeans components, J2EE Connector Architecture, J2EE Management Model, J2EE Deployment API, Java Management Extensions (JMX), J2EE Authorization Contract for Containers, Java API for XML Registries (JAXR), Java Message Service (JMS), Java Naming and Directory Interface (JNDI), Java Transaction API (JTA), CORBA, and JDBC data access API.

What's more, the Java 2 Platform, Enterprise Edition, takes advantage of many features of the Java 2 Platform, Standard Edition, such as "Write Once, Run Anywhere" portability, JDBC API for database access, CORBA technology for interaction with

existing enterprise resources, and a security model that protects data even in internet applications. Building on this base, Java 2 Enterprise Edition adds full support for Enterprise JavaBeans components, Java Servlets API, JavaServer Pages and XML technology. The J2EE standard includes complete specifications and compliance tests to ensure portability of applications across the wide range of existing enterprise systems capable of supporting J2EE.

2.2 Java 2 Platform, Micro Edition (J2SE) Overview

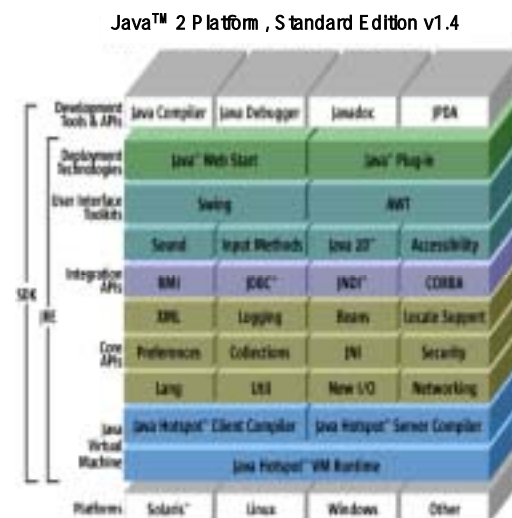
The J2SE platform is a fast and secure foundation for building and deploying client-side enterprise applications. In today's .com world of nanosecond response times and information gratification, J2SE technology provides the speedy performance and high functionality that is demanded by Web users.

For end users, J2SE technology enables faster and easier use of functionally rich Web applications, such as corporate intranets and interactive shopping aids for eCommerce. For enterprise developers, the improved J2SE technology serves as the base tool for creating sophisticated, valuable applications that can be brought to market quickly.

There are two principal products in the J2SE platform family: Java 2 Runtime Environment, standard Edition (JRE) and Java 2 Software Development Kit, Standard Edition (SDK). The JRE provides the Java APIs, Java virtual machine, and other components necessary to run applets and applications written in the Java programming language. It is also the foundation for the technologies in the Java 2 Platform, Enterprise Edition (J2EE) for enterprise software development and deployment. The JRE does not contain tools and utilities such as compilers or debuggers for developing applets and applications.

The Java 2 SDK is a superset of the JRE, and contains everything that is in the JRE, plus tools such as the compilers and debuggers necessary for developing applets and applications.

This conceptual diagram illustrates all the component technologies in J2SE platform and how they fit together.



The J2SE platform works with an array of tools, including Integrated Development Environments (IDEs), performance and testing tools, and performance monitoring tools.

2.3 Java 2 Platform, Micro Edition (J2ME) Overview

The Java™ 2 Platform, Micro Edition (J2ME™) is the Java platform for consumer and embedded devices such as mobile phones, PDAs, TV set-top boxes, in-vehicle telematics systems, and a broad range of embedded devices.

Like its enterprise (J2EE™), desktop (J2SE™) and smart card (Java Card™) counterparts, the J2ME platform is a set of standard Java APIs defined through the Java Community Process program by expert groups that include leading device manufacturers, software vendors and service providers.

The J2ME platform delivers the power and benefits of Java technology tailored for consumer and embedded devices — including a flexible user interface, robust security model, broad range of built-in network protocols, and support for networked and disconnected applications. With J2ME, applications are written once for a wide range of devices, are downloaded dynamically, and leverage each device's native capabilities.

The J2ME platform is deployed on millions of devices — from mobile phones, to PDAs, to automotive devices — supported by leading Java technology tools vendors, and used by companies worldwide. In short, it is the platform of choice for today's consumer and embedded devices.

The J2ME architecture can be seen from the following figure.



3. THE APPLICATION OF JAVA PLATFORM IN WEB-BASED SIMULATION AND COMPUTING

3.1 The Application in Web-based Simulation

The Mars Simulation Project [2] is a free software Java project to create a simulation of future human settlement of Mars.

The simulation is a multi-agent artificial society set in a detailed virtual world. The programming is object-oriented, with everything in the virtual Mars, including Mars itself, modeled as interacting objects.

The settler AI is based on performing missions and tasks. These are determined randomly by likely hood and bound by situational limits. Every simulation run produces different results.

XML configuration files allow the user to modify the simulation

properties.

3.2 The Application in Distributed computing

A simple programmable Java platform-independent distributed computation system has been developed to exploit the free resources on computers linked together by a network. It is a multi-tiered distributed system model, which is unbounded in principal. The system consists of an n-ary tree of nodes where the internal nodes perform the scheduling and the leaves do the processing. The scheduler nodes communicate in a peer-to-peer manner and the processing nodes operate in a strictly client-server manner with their respective scheduler. The independent schedulers on each tier dynamically allocate resources between jobs based on the constantly changing characteristics of the underlying network.

The foundations for the multi-tiered distributed computation system were laid in the Java Distributed Computation Library (JDCL) [4] and its extensions [5], which provide an emulated MIMD pipeline processor. The JDCL provided a simple development platform for developers who wished to quickly implement a distributed computation system. It arose out of the need for a platform-independent distributed system that was easy to create, adapted to system changes, and was easy to deploy. Systems such as SETI@home did not address these issues very well and were designed to be platform dependant and for a single purpose only. The JDCL does, however, surer from similar scalability problems to those of SETI@home in that it has one server (single machine or cluster). The design of the current multi-tiered system aims to address this concern.

3.3 The Application of Java Platform in Mobile Computing

Java Based Integrated Application Development Tools On server Side - Java has become a standard dominant language for server-side programming. Java makes it easier to write safe, reliable code through features, such as automatic memory management and structured exception handling. A large set of APIs and cross-platform design provide power and portability. Sun has announced significant enhancements for mobile computing and interfaces to wireless networks. Several application servers support Java interfaces.

For example, J-Phone Java-based mobile phone [3] use the Java 2 platform, micro edition (J2ME) .It employs the mobile information device profile, an industry standard, for the specifications called profile, which is used to develop Java applications. And it also supports for MIDP, which is capable of running the same applications regardless of the communication method device. So the J-Phone Java-based mobile phone has many advantages to the Java-based content and services. For example, it's agent software for automatically obtaining data can be enhanced from networks; it has better gaming software; and it has been improved security during time spent accessing content.

4. CONCLUSIONS

Except for the things described above, there are many other Java technology and applications, such as Java Card Technology and applications in web services and so on. Generally speaking, Java technology gives us a platform-independent way of doing things.

And there will be more and more the applications of Java platform in Web-based simulation and computing.

5. REFERENCES

- [1] <http://java.sun.com>
- [2] <http://mars-sim.sourceforge.net>
- [3] <http://www.mobileinfo.com>
- [4] K. Fritsche, J. Power, and J. Waldron. A Java distributed computing library. In 2nd International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT2001), pages 236–243, Taipei, Taiwan, July 2001.
- [5] T. Keane, R. Allen, T. J. Naughton, J. McInerney, and J. Waldron. Distributed Java platform with programmable MIMD capabilities. In N. Guelfi, E. Astesiano, and G. Reggio, editors, Scientific Engineering for Distributed Java Applications, volume 2604 of Springer Lecture Notes in Computer Science, pages 122–131, Feb. 2003.

Semantic Web Enabled the Context Information in Ubiquitous Computing System *

Chen Xuhui^{1,2} Tang Shancheng² Wang Yimin¹

School of Computer and Communication, Lanzhou University of Technology¹
Lanzhou, Gansu 730050, China

Compute Information and Technology Institute, Xi'an Jiaotong University²
Xi'an, Shaanxi 710049, China.

Email: xuhui.chen@163.com Tel.: +86-931-2973900

ABSTRACT

In Ambient intelligence environment, the surrounding and the available information is hoped to be found, utilized and reacted actively, thus improved the distributed human-machine interaction ways greatly, in other words, the application of context aware technology in ubiquitous computing system has great application scene. In this paper, we developed an Intelligent environment called the personified home service system, which we have implemented using standard Semantic Web (RDF, OWL, DAML), Web Services (SOAP, WSDL) and pervasive computing (UPNP) technologies. Extending the human-machine interaction, home devices such as sensor, TV and refrigerator could be used as interactive device not only Mouse and CRT. It offers an incentive to device manufacturers to incorporate semantic web technologies into their devices in order to get the benefits of easier and more flexible use of their devices' features by end-users. For extensive intelligence in the system, the Semantic Web can assist the evolution of human knowledge as a whole. We analyze user's daily record and predict the user's interest, and find user's potential interests through feedbacks. The Semantic Webs will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated tasks for users.

Keywords: Semantic Web, Ubiquitous Computing, Context Aware, Human Machine Interaction, and Agent.

1. INTRODUCTION

As Computing becomes more pervasive, the nature of applications must change accordingly. In particular, applications must become more flexible in order to respond to highly dynamic computing environments, and more autonomous, to reflect the growing ratio of applications to users and corresponding decline in the attention a user can devote to each. That is, applications must become more context-aware. To facilitate the programming of such applications, infrastructure is required to gather, manage, and disseminate context information to applications [1]. The semantic web is enabled the context information in ubiquitous computing system.

The semantic Webs, a new form of Web content that is meaningful to computers, will unleash a revolution of possibilities, not only can be used to browse but also can enable machines to comprehend people's thoughts by semantic documents, not just by human speech and writings. The Semantic Webs will bring structure to the meaningful content of Web pages, creating an environment where software agents roaming from page to page can readily carry out sophisticated

tasks for users [2]. Two important technologies for developing the Semantic Web are already in place: extensible Markup Language (XML) and the Ontology.

In addition, if properly designed, the Semantic Web can assist the evolution of human knowledge as a whole. We studied context aware application in Personified Home Service System, aiming at building up a natural, kindly and vivid home service system. In personified home service system, we abandon the idea of a single concentrated interface such as CRT and keyboards, and instead treat the whole environment (such as a home) as the interface, seamlessly and immediately. Appliances, screens, chairs and etc can all act as input and an output device, but also the personified home itself is a distributed system of agents responding to our actions. In the age of context aware application, the human-machine interaction interface should be distributed and ubiquitous. The Semantic Web will break out of the virtual realm and extend into our physical world, and URLs can link to anything including physical entities, which mean we can use the RDF language to describe devices such as cell phones, TVs, refrigerators, air-conditions. Such devices can advertise their functionality what they can do and how they are controlled through software agents. Being more flexible than low-level protocols such as UPNP, JINI, this semantic approach opens up a world of exciting possibilities.

2. THE SEMANTIC WEB ENABLED THE UBIQUITOUS APPLICATION.

Ubiquitous environment can be found in different literatures with different definitions. Gregory D. Abowd, Georgia Institute of Technology defines context aware computing as work that leads to the automation of a software system based on knowledge of a user's physical, social, emotional, or informational state [3]. We believe that a user's context should basically include these various kinds of information and the relations of the user: the user's physical environment, such as social relations, social position, and group relations; the user's psychological status, such as emotional state, happiness /unhappiness and whether they are busy, etc. How to describe and predict these various kinds of information and the relations? In this paper, we have implemented using standard Semantic Web (RDF, OWL, DAML-S), Web Services (SOAP, WSDL) and pervasive computing (UPNP) technologies.

2.1 Scheduling a Future Presentation of Life

The IST Advisory Group (ISTAG) was to describe what living with 'Ambient Intelligence' might be like for ordinary people in 2010. Four Scenarios are not traditional extrapolations from the present, but offer provocative glimpses of futures that can (but need not) be realized. [4]

We envision an active smart home to contain hundreds, or even thousands, of devices and sensors that will be everywhere, performing regular tasks, providing new functionality, bridging the virtual and physical worlds, and allowing people to communicate more effectively and interact seamlessly with available disappearing computing resources and the surrounding physical environment.

2.1.1 The Aim of a Future Home

The main feature of the personified home enabled system with respect to its relation to the persons using it can be characterized by:

- Non-obtrusive: many often-invisible distributed devices exist embedded in the environment, not intruding upon our consciousness, unless we need them
- Personalized: its behavior can be tailored towards personal needs and can recognize the user.
- Adaptive: its behavior can change in response to a person's actions and environment.
- Anticipatory: it anticipates a person's desires and environment as far as possible without conscious mediation. The human is the center of the home, fulfills the characteristic of the personal life and setups the natural, kindly and vivid home service system.

2.1.2 Scheduling of a Future Life

It is six o'clock in the afternoon. Jim and his wife Mary are taking a coffee in the home, and don't want to be excessively bothered during this pause. They are proud of their affective computing devices such as PDA. It can record the person's hobby, interest, calendar and so on. Nevertheless, all the time they are receiving and dealing with incoming calls and mails. The entertainment system was belting out the MTV "Say you say me" when the phone rang. Bob, Jim's friend, invite him to play tennis these days. PDA agent will consult with his friends by E-mail to arrange the proper time and then book the tennis court. This information is shared with the Bob's PDA, not with the Bob himself as to avoid useless information overload. Meanwhile, a call from Lucy, Mary's mother, is further analyzed by her PDA. When Mary answered, his phone turned the sound down by sending a message to all the other local devices that had a volume control. Lucy was on the line from the doctor's office: "I needs to see a specialist and then has to have a series of physical therapy sessions. I'm going to have my agent set up the appointments." ...

Jim and Mary could use their soft agents of PDA to carry out all these tasks thanks not to the World Wide Web of today but rather the Semantic Web that it will evolve into tomorrow.[5]

2.2 Infrastructure of Ubiquitous Computing

Currently, the programming of ubiquitous computing applications is complex and laborious. This situation could be remedied by the creation of an appropriate infrastructure that facilitates a variety of common tasks related to context awareness, such as modeling and management of context information. Semantic Web and Web Services technologies can meet in ubiquitous computing environment

2.2.1 Semantic Infrastructure for Ubiquitous Computing (SIUC)

In this section, we describe how Semantic Infrastructure for Ubiquitous Computing (SIUC) works in the future life. SIUC offers a user interface for presenting to the user what she can do in the current context, and lets her define and execute the

tasks.

First there is a discovery phase. SIUC searches for the local and discovery mechanism (UPnP) available and keep checking newly added or removed services. SIUC consults the local service manager (see Figure 1) for the available local semantic services such as "Local File" service, which lets you choose a local file and expose it as a Web page. For each of those local services, STEER retrieves its Semantic Service Description (SSD) in OWL+DAML from the local file system and feeds it into its inference engine (IE).

Seconds, SIUC searches the services on the network using UPnP device discovery protocol. The UPnP architecture offers pervasive peer-to-peer network connectivity of devices and PCs. For each of those found UPnP devices, SIUC makes a specific UPnP call (getDescriptionURL (OWL+DAML)) into it to get the URL of its SSD (expressed in). SIUC determines the service as a semantic service if it gets a result for this UPnP call. Then, SIUC downloads its SSD in DAML from the URL returned. SIUC feeds the SSD into its IE.

For device manufacturers, devices can be treated as Semantically Services Described (SSD); someone will have to provide this semantic layer wrapper of the devices' functionality. In addition, the devices need to implement a service discovery mechanism (e.g., UPnP or Bluetooth) and make available a remote control API for accessing their functionality (e.g., UPnP). Device manufactures will benefit from the newly found uses of their devices in flexible and ad-hoc Task Computing environment. [6][7]

2.2.2 RDF of Home Devices

All home services, such as television or telephone, serve for different family members in one's own way. The member's interests and agenda determine what kinds of service will be provided and how home devices will work. Those devices work together smoothly depending on their RDF schemas. Figure 2 shows structure of home devices' RDF schemas.

Devices descriptions are documented in standard RDFS language. The elements of a RDFS document are: device Type represents device's classification which is used to distinguish from others; device Name; device ID stands for the unique ID of device in home network; device Security Class; service List display all the services this device can provide; service Type serve as service's classification which is used distinguish from other services; service URL links the document of service description.

By usage, home devices are classified as: mobile network devices (Pocket PC, Smart Phone or etc), accessorial devices (printer, projection, whiteboard), digital entertainment devices (MP3 player, digital video), home appliances (television, air-condition, washer, refrigerator and microwave oven), control devices (input circuit, executor, sensor and other unit).

Services descriptions are documented in standard RDF language, which is a part of Web Services Description Language. The document's root element is '<definitions>' which includes many child elements. And these child elements are basically classified as two classes: the former exist in the front part of document and are made up of service's 'Abstract

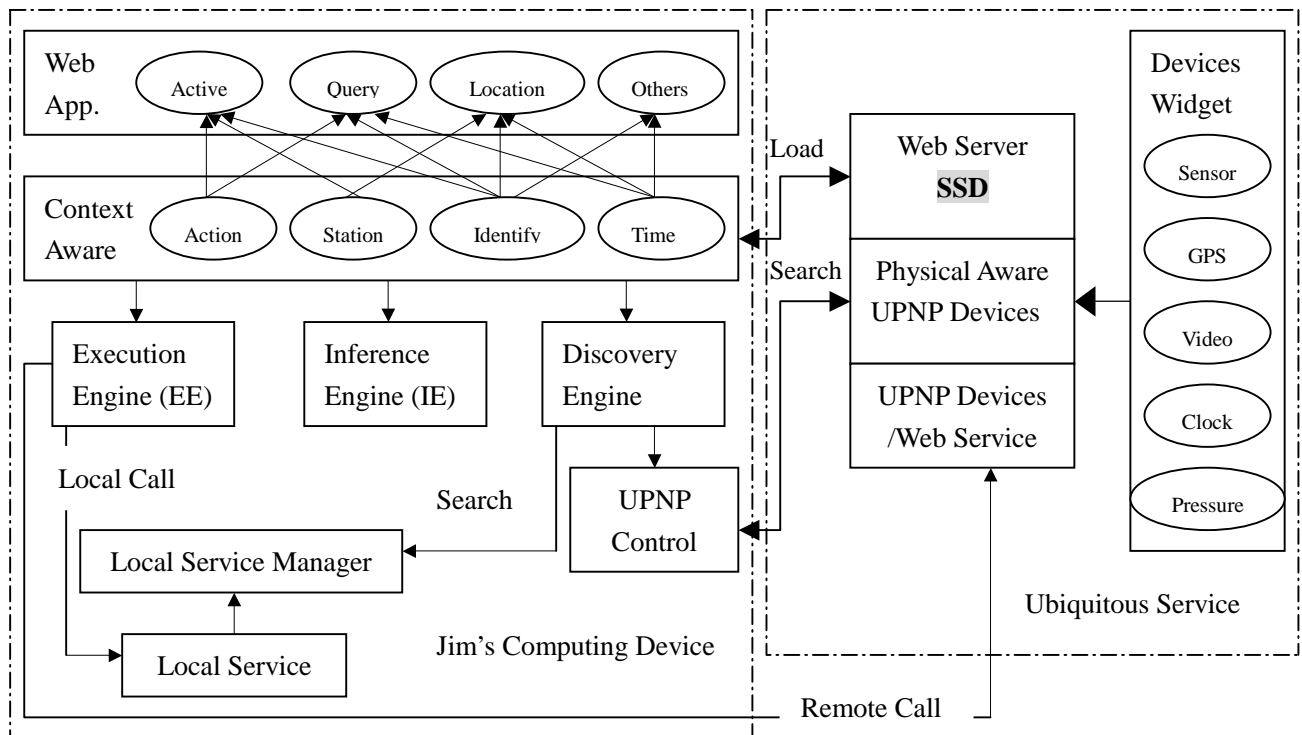


Figure 1 Semantic Infrastructure for Ubiquitous Computing

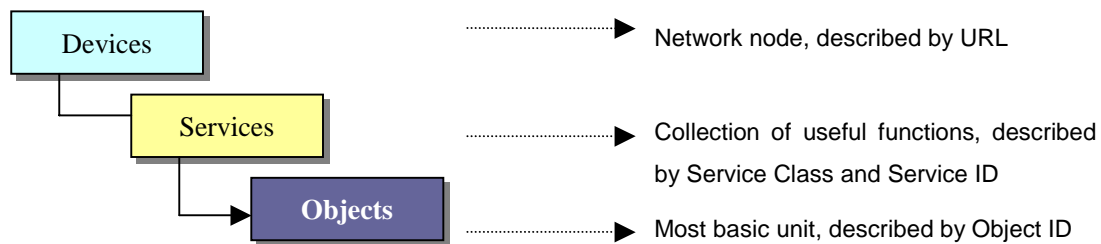


Figure 2 structures of home devices' RDF



Figure 3 Smart Agent Based on Semantic Web

Definition'; the latter exist in the back part of document and are formed into service's 'Concrete Illumination'. Abstract definition describes a service supporting any operation system and any program language, includes three elements: type, message and port Type; Concrete illumination details a particular service and includes two elements: binding and service.

3. SMART AGENT EXCHANGE THE PROOFS OF SEMANTIC WEB

The real power of the Semantic Web will be realized when people create many applications that collect Web contents from diverse sources, process these information and exchange the results with other applications. Software agents will be greatly facilitated by semantic contents on the Web. The effectiveness of such software agents will increase exponentially as more machine-readable Web contents and automated services (including other agents) become available. For extensive intelligence in the system, the Semantic Web can assist the evolution of human knowledge as a whole. We analyze user's daily record and predict the user's interest, and find user's potential interests through feedbacks.[8]

3.1 Intention Agent Feedbacks on Personal Hobby and Interests

Personal resource description frame establish a calendar for every user. User's interests are record in his calendar. One user may have many interests in a period, and then a calendar is made of many personal interests, which are represented by 3-Descriptor model (Positive, Negative and Long-term descriptor). Long-term descriptor stand for user's lasting interest and is modified stepwise by user's long-term behavior; Positive descriptor and negative descriptor serve as user's short-time interest, and are adjusted rapidly by user's operations. Positive descriptor describes user's short-time interests according with user's fresh requirement and plan, while negative descriptor represents what user feel no interest in.

User personal model described by RDF makes it possible to predict people's interests. User's feedbacks show what user's interests are and the grade of interests. Through feedbacks, system can get unknown knowledge and find user's potential interests.

An feedback is composed of four elements: the type of feedback, the contents on which feedback is received (such as audio, video, document and etc), the degree of interest, operation frequency.

According to what user access information and services by means of, there are two kinds of feedbacks: direct and indirect feedbacks. The value of feedbacks is either positive or negative. Feedback is positive means that user was interest in the contents that system provided, while negative means the not. The degree of interests show how much does feedback has an effect on interest, and it's value is depended on whether user is satisfied or not.

Feedbacks can be used to establish heuristic rules. Feedbacks can be either direct or indirect. User himself provides direct feedback. User inputs his interests into system at first hand. After received feedbacks, agent gets some user's interests and modifies user's calendar. Direct feedback has both advantages

and disadvantages. Its advantages are obvious: feedbacks can be exactly right and agents can know user's interests directly. Disadvantages are: direct feedbacks need to be provided by user and user may be tired of this.

Indirect feedbacks are obtained from system, such as from playing films or music, or from looking over documents. Because agent predicts user's interests through indirect feedbacks, it is not so precisely as direct feedbacks. For example: in a recommendation of music list provided random by system, user choose one and play it. That means user may be interest in this music and then personal services will collect singer files, comments and other likely music for further requirements. Otherwise, if some kinds of music haven't been played for a long time, then user may have no interest in it. By this way, system predicts user's interest indirectly, so it is not so precisely as direct feedback. But indirect feedback can be endowed with a low degree of interests in order to avoid that a wrong prediction has a bad effect on user's calendar.

Therefore, State and smart action of the intention agent can be described as:

$$Agent_Model_{Ai} = \langle D, K, I, Type, Positive, Negative \text{ and } Long\text{-}term\ descriptor \rangle$$

D, K, I means Database, Knowledge base and Intention base.

3.2 Intention Agent Based on Semantic Web

The Semantic Web promotes this synergy: even agents that were not expressly designed to work together can transfer data among themselves when the data come with semantics.

We advocate the agent coordination mechanisms in our software. The software system is Client/Server structure to implement resource agents, task agents, and manage agents. Make no mistake: to create complicated value chains automatically on demand, some agents will exploit artificial intelligence technologies in addition to the Semantic Web. The consumer and producer agents can reach a shared understanding by exchanging ontology, which provide the vocabulary needed for discussion. Agents can even "bootstrap" new reasoning capabilities when they discover new ontology.

For instance, Semantic descriptions of devices capabilities and functionality RDF will let us achieve such automation with minimal human intervention. We can use the RDF language to describe an academic conference, daily management, hobby and interest, and home devices. Such descriptions can advertise their functionality that they can do, how they are controlled, and which attribution they are owned as well as software agent. A trivial example occurs when Jim is working in the office and is invited by an academic conference from conference's task agent, whose conference topics are very interesting to Jim. Jim's manage agent could interact with conference's task agent, and then, his manager sends this information to his task agent and resource agent in order to modify his calendar. The query agent adopts Jena systematic RDF inquiry language RDQL.[10]

RDQL can filter out the contents that the user isn't interested in, and this can increase search efficiency greatly. This typical process will involve the creation of a "value chain" in which subassemblies of information are passed from one agent to another, each one "adding value," to construct the final product requested by the end users.

4. CONCLUSIONS

In this paper, we studied context aware application in Personified Home Service System, aiming at building up a natural, kindly and vivid home service system. We abandon the idea of a single concentrated interface such as CRT and keyboards, and instead treat the whole environment (such as a home) as the interface, seamlessly and immediately. The Semantic Web will break out of the virtual realm and extend into our physical world, and URLs can link to anything including physical entities, which mean we can use the RDF language to describe devices such as cell phones, TVs, refrigerators, air-conditions. Being more flexible than low-level protocols such as UPNP, JINI, this semantic approach opens up a world of exciting possibilities.

We see Ubiquitous Computing system as a business opportunity for both device manufacturers and IT solution providers. The researcher should explore the characteristics of context information in ubiquitous systems and describe a set of context modeling concepts designed to accommodate these.

5. REFERENCES

- [1] Karen Henricksen, Jadwiga Indulska, Andry R., Pervasive 2002, LNCS 2414, Springer-Verlag Berlin Heidelberg 2002, pp. 167-180.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001.
- [3] A.K. Dey, D. Salber, and G.D. Abowd, Context based Infrastructure for Smart Environments, Proc. 1st Int'l Workshop on Managing Interactions in Smart Environments (MANSE 99), Springer-Verlag, New York, 1999, pp. 114-128.
- [4] K. Ducatel, M. Bogdanowicz, Scenarios for Ambient Intelligence in 2010, IPTS-Seville, Feb. 2001, pp4-21.
- [5] Chad Burkey, Environmental Interfaces: HomeLab, 2000 Conference on Human Factors in Computer Systems (CHI 2000), pp1-2.
- [6] Resource Description Framework (RDF) Schema Specification 1.0, W3C Recommendation 27 March 2000, <http://www.w3.org/TR>
- [7] Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies W3C Working Draft 2003, <http://www.w3.org/TR/2000/>
- [8] Terry R. Payne, Rahul Singh and Katia Sycara, Calendar Agents on the Semantic Web, IEEE Intelligent Systems, Vol. 17(3), May/June 2002, pp84-86.
- [9] Kovacs E, Rohrlé K., Schiemann B, Adaptive mobile access to context aware services, Agent Systems and Applications, 1999 and Third International Symposium on Mobile Agents. Proceedings. First International Symposium on, 1999 pp 190-201.
- [10] Retsina semantic web calendar agent, <http://www.daml.ri.cmu.edu/Cal/>
- [11] Li Weihua, Ontology Supported Intelligent Information Agent, 2002 First International IEEE Intelligent System, Sept. 2002, pp383-387.



Chen Xuhui is a vice Professor of Lanzhou university of technology. He received the PhD of Computer Science and Technology Institute from Xi'an Jiaotong University, People's Republic of China, His research interests include context aware computing, wearable computing, active knowledge base, intelligent agents, and human interaction.

Contact him at College of Electrical and Information Engineering, Lanzhou University of Technology, China. Email: xuhui.chen@163.com.

Tang Shancheng is a PhD student of Computer and Information Technology Institute, Xi'an Jiaotong University, People's Republic of China, His research interests include Ad hoc network, pervasive computing, active knowledge base, ambient intelligent, and gesture recognition. Contact him at Computer and Information Technology Institute, Xi'an Jiaotong University.

Wang Yimin, The School Dean of Computer and Communication, Lanzhou University of technology.

A Web-based Engineering Optimization System and Its Application

Caijun Xue, Hong Nie, Yanqin Dai

College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics
Nanjing, 210016, P.R.China

Qingying Qiu

State Key Laboratory of CAD&CG, Zhejiang University
HangZhou, 310027, P.R.China

ABSTRACT

Web-based computing technology has been popular in the engineering fields since it is promising to use computing resources more efficiently. The authors have developed a methodology that takes advantages of the World Wide Web to realize design optimization of an engineering system. A web-based optimization system framework is constructed. JSP/Serverlet technology is used for web server implementation. JAVA JNI programming realizes a link between the serverlets and the optimizers. JAVA JDBC programming realizes to access optimization models in the SQL database server. This paper describes these methodologies in detail. The design optimization of working equipment of a hydraulic excavator is used to illustrate the application of the web-based optimization system.

Keywords: Engineering optimization, Web-based computation, JAVA programming, JSP/Serverlet

1. INTRODUCTION

The engineering optimization technique is one of most important methods to the product developing engineers. However, development of a large-scale optimization system is usually beyond their abilities. Moreover, much commercial software is expensive to buy for small companies and require lengthy education of the operators. As a result, many practical engineering problems can't be designed by any optimization tools. Therefore, any inexpensive and easy-to-use engineering optimization tools are very welcoming.

The web-based computing concept is a popular topic of discussion among users in the computer industry. Among other well-known advantages such as accessibility, the World Wide Web offers many new informational technologies and distributed computing [1]. The distributed computing concept means that instead of having all the computation taking place on a user's local machine, the user's computer sends data to a powerful remote server, which returns results which are then displayed on the local machine. A Web interface is software which serves as a bridge between the client's computer and the remote server. Thus, the web-based technology is actively being introduced to the engineering optimization field [2] [3]. The web interface provides a simple way to the engineering optimization software package.

Our present project is another attempt to help in improving the situation by developing a web-based engineering optimization system. In our case this means that a client, even in a platform with relatively low computing ability, can run engineering optimization problems without any special software installed but a Web browser. The paper is built up as follows. Section 2 is a brief introduction to the web optimization system. Section 3 discusses technical details of our implementation of the web-based engineering optimization system. Section 4 describes the applying method of the system by an example. Finally, conclusion will be drawn.

2. WEB-BASED OPTIMIZATION SYSTEM

The framework of the web-based optimization system is designed as shown in Fig.1, including four main components. They are HTTP clients, a web server, modeling interfaces and an application server. These components may be located in different geographical places but all are linked through the internet.

The application server provides powerful optimization engine, high-end computers, optimization models, and paralleled computing environment. The web server acts as a bridge between the application server and the HTTP client. The remote designers can browse the information of optimization models in the application server, plan optimization scheme, start complex engineering optimization process, and get optimization result through the web sever. The web server also acts as a bridge between the application server and the modeling interface. The modeling users can finish modeling by manipulating the optimization models database and testing the models by using the optimization computing circumstance. The optimization circumstance is a multi-agent based distributed computing circumstance as discussed in ref. [1]. All distributed computing resources (including personal computers, SGI workstations, and SUN workstations) are integrated by network programming. An optimization controller is linked to a distributed optimization system based multi-agent technology [4]. Several practical optimization algorithms are provided in the distributed optimization circumstance. These algorithms are programming by C++ language and be linked to the distributed through JNI programming [5].

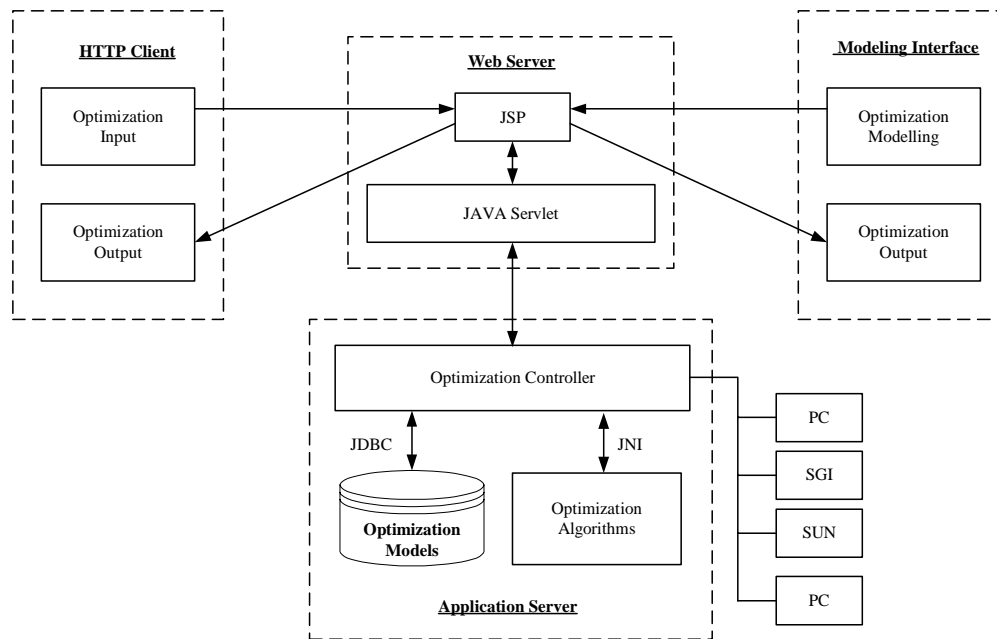


Fig.1 Framework of the web-based optimization system

3. SYSTEM IMPLEMENTATION

Central parts of the framework are a java web severer based on JSP/Serverle. A web application is a collection of servlets, HTML pages, classes, and other resources that can be bundled and run on multiple containers from multiple vendors. Tomcat 4.0, which is an open-source Java-based web application container created to run servlet and JSP Web applications, is used in this paper to realize a web application. In our framework, the servlets implement a service that mediates the data between the users and the optimization controller.

As discussed in part 2, the application server is the most important component of the web-base optimization system. Java is used to develop the system, as it is a robust and reliable programming language that is platform independent and Internet supported. It is the best choice for developing a distributed optimization system running on the Internet. All optimization models are centrally managed by a SQL database management system (DBMS). In order to build, modify and invoke optimization models in java language, a java-based database access method is implemented through DBMS JDBC driver, which is a standard interface which allows Java applets and applications to inter-operate with database.

4. ENGINEERING APPLICATIONS

Objective

With the tendency towards high speed and efficiency, the working conditions of the modern heavy machine tends to be poor, time-varying with frequently accompanying impact, the arm structures of the construction machinery tend to be larger and heavier than the current type. Such a design policy causes increasing complex design requirements, such as lower manufacturing cost, greater digging range, lower vibration and noise, lower exhaust air pollution and higher durability, etc. For example, design of hydraulic excavator working

equipment consists of boom, arm, bucket, guide linkage, bucket linkage, boom cylinder, arm cylinder and bucket cylinder. The design process needs the coordination of engineers with different disciplinary knowledge. So, it is very promising to provide some easy-use tools to realize design optimization of hydraulic excavator working equipment. These tools provide some friendly user interfaces, including an optimization modeling interface, an optimization control interface and an optimization result interface.

Optimization modeling

Optimization modeling interface is relatively simple which is shown as Fig.2. Firstly, an optimization model name should be assigned in the first line, and then modeling users can define design variables and design functions according to the restricted format shown by the modeling example. When finishing modeling, left-clicking the "finished" button will send the model information to the web server. A related serverlet will write them to the model database. 43 variables and 112 design functions is included in the optimization model of hydraulic excavator working equipment.

Optimization control

Optimization control interface is java applet shown in Fig.3. This applet includes a text box, a text field, two table, and several labels and buttons. It is designed to browser, modify and build optimization models. After a database server name and a database name are selected, optimization models are listed in the text field. The users can select a model to browser its information. Then, the users can quickly build a new model based on the present model in the database by modify variable names, variable initial value, variable low limit, variable upper limit, variable names, variable initial value, variable low limit, variable upper limit. Some new variables and function can be added to the new model, and some unimportant variables and function can be removed from the new model. In the end, left-clicking the "confirm" button will send an optimization computing demand to the web server.

Optimization results

After the optimization result is got, the web server sends it to the user by a web page. Optimization result interface is shown in fig.4. This page includes two forms and an applet. One form lists the information of variables, and the other lists the

information of objective. Additionally, the applet can provide users more information about design variables and design functions. You can select a variable or function to view its optimization history in a graphic format.

MODELING INTERFACE OF THE WEB-BASED ENGINEERING OPTIMIZATION SYSTEM

modeling help

First you need to define all the design variables. And then you can model your problem only with these variables and constant value. When finishing it, click the button named "finished".

Input an model name:

Define design variables

Name	Low limit	Upper limit	Initialized value
For example:			
x1	5.1	5.5	5.2
x2	5.1	5.5	5.2
x3	5.1	5.5	5.2
x4	5.1	5.5	5.2
x5	5.1	5.5	5.2

Define design objective in java language

For example:
`fun=3*x1*x2*x3*x4*x5+123;`

Define design constraints in java language

Expression	Low limit	Upper limit
For example: <code>con=3*x1*x2*x3*x4*x5+123;</code>	0	100.0

Fig.2 Optimization modeling interface

CONTROL INTERFACE OF THE WEB-BASED ENGINEERING OPTIMIZATION SYSTEM

Server:

Database:

Models:

- Model01
- Model02
- Model03
- X0001
- X0002
- X0003
- WorkEquipment01
- WorkEquipment02
- WorkEquipment03
- Test01
- Test02
- Test03
- Test04
- Test05
- Test06
- Test07
- Test08

Design Variables

VarNo	Var name	Description	InitValue	Low Limit	Upper Limit	Coef
1	x1	x1	2500.0	2000.0	3000.0	
2	x4	x4	1423.8	1200.0	1500.0	
3	x5	x5	2415.8	1600.0	4200.0	
4	x6	x6	1345.5	1100.0	1400.0	
5	x7	x7	1345.5	1300.0	1600.0	
6	x8	x8	915.8	850.0	1000.0	
7	x9	x9	1475.8	1350.0	1600.0	
8	x10	x10	206.8	270.0	500.0	
9	x11	x11	1333.5	1600.0	1400.0	
10	x12	x12	306.8	280.0	450.0	
11	x13	x13	928.8	850.0	1000.0	
12	x14	x14	1436.8	1450.0	1600.0	

Design Functions

Fun No	Fun Name	Description	Old Low Limit	Old Upper Limit	Old Weight	Coef Low
1	Fun01	Fun01	4.0	12.0	0.0	
2	Fun01	Fun01	0.0	10000.0	0.1	
3	Fun02	Fun02	0.0	10000.0	0.1	
4	Fun03	Fun03	0.0	10000.0	0.1	
5	Fun01	Fun01	0.1	10000.0	0.0	
6	Fun02	Fun02	0.1	10000.0	0.0	
7	Fun03	Fun03	0.1	10000.0	0.0	
8	Fun04	Fun04	0.1	10000.0	0.0	
9	Fun05	Fun05	0.1	10000.0	0.0	
10	Fun07	Fun07	0.1	10000.0	0.0	
11	Fun12	Fun12	0.0	600.0	0.0	

Fig.3 Optimization control interface

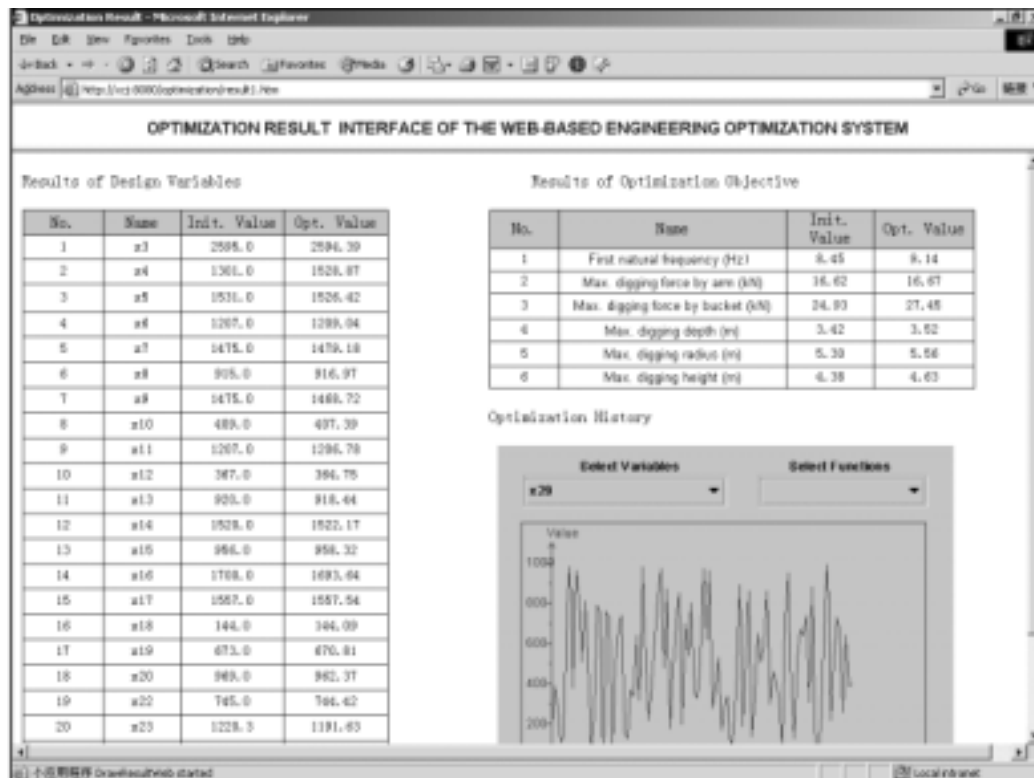


Fig.3 Optimization result interface

5. CONCLUSION

A web-based engineering optimization system is introduced in this paper. It facilitates the application of engineering optimization technology by integrating web service software in the optimization system. The proposed system has following features: (1) Easy access through web browsers, and make optimization technology available to a broader range of users and applications. (2) Share powerful optimization engine, high-end computers, optimization models, and paralleled computing environment. (3) Easy support and maintenance, and Better technology security

6. REFERENCES

- [1]. Kiyoko F., D. T. LEE. "Towards web-based computing", International Journal of Computational Geometry & Applications 11 (1), 2001, 71-104.
- [2]. D. Tcherniak and O. Sigmund. "A web-based topology optimization program", Structural and Multidisciplinary Optimization 22 (3), 2001, pp. 179-187.
- [3]. Xavier Drèze, Fred Zufryden, "A Web-Based Methodology for Product Design Evaluation and Optimization", Journal of the Operation Research Society, Vol. 49, 1998.
- [4]. Caijun Xue, Qingying Qiu, Peien Feng et al. "Research on distributed collaborative optimization technology", Proceedings of 2002 international symposium on distributed computing and applications to business, engineering and science. Wuxi, Jiangsu, P.R. China, 2002:138-143.
- [5]. Bruce Eckel. Thinking in Java (2nd Edition). Beijing: China Machine Press, 2002.



Caijun Xue is a postdoctoral fellow in Aircraft Design Department, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. He has published 10 Journal papers. His research interests are distributed parallel and multidisciplinary design optimization.



Hong Nie is a Full Professor in Aircraft Design Department, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics. He has published two books, over 50 Journal papers. His research interests are computer-aided design, computer-aided engineering, distributed parallel in engineering, multidisciplinary design optimization of landing gears, and advanced aircraft design technology.

Research on the Model of Intelligent Meta-search Engine

Li Liu and Wenbo Xu

College of Information Technology, Southern Yangtze University, Wuxi, Jiangsu, 214000, China

Email: wxliuli@sytu.edu.cn, Email: xwb@sytu.edu.cn

ABSTRACT

The amount of information available via networks and databases has rapidly increased and continues to increase. Existing search and retrieval engines provide limited assistance to users in locating the relevant information that they need. Intelligent Meta-search Engine may prove to be the needed item in transforming passive search and retrieval engines into active, personal assistants. This proposal explores the quantity of information available that is driving the need for improved search and retrieval engines. It then reviews current information retrieval literature and agency literature. Following these reviews, it proposes that the combination of effective information retrieval techniques, meta-search engine and autonomous, intelligent agents can improve the performance of information retrieval in an existing search or retrieval engine. The components model of Intelligent Meta-Search Engine is also given. The proposal then presents the performance evaluating of this research and the methodology to build these component.

Keywords: Intelligent agent, Meta-search Engine, Data mining, Decision tree.

1. INTRODUCTION

The Web organizes information by employing a hypertext paradigm. Users can explore information by selecting hypertext links to other information. As the Web continues its explosive growth, the need for searching tools to access the Web is increasing. Yahoo! is the big name in Web directories [1]. Examples include Alta Vista, InfoSeek, Open Text and Excite. However, these search engines are not as sophisticated as one might expect.

Meta-Search Engine is the technology to enhance search engine performance. It works when receiving user's search inquisition, formats them and submits them to independent search engines that are selected with a set of pre-determined criteria. The results from these independent search engines will be analyzed, integrated and resorted according to relevance before being sent back to the user. Obviously, the advantage of it is to simplify search work from surfing in the Internet, as well as to merge the advantages of these independent search engines.

By using a meta-search engine, you get a snapshot of the top results from a variety of search engines (including a variety of types of search engines), providing you with a good idea of what kind of information is available.

Meta-search engines are tolerant of imprecise search terms or inexact use of operators, and tend to return fewer results, but with a greater degree of relevance. They're best to use when you've got a general search, and don't know where to start - by providing you results from a series of sites, they help you to determine where to continue focusing your efforts (if this

proves necessary). They also allow you to compare what kinds of results are available on different engine types (indexes, directories, pay-for-placement, etc), or to verify that you haven't missed a great resource provided by another site, other than your favorite search engine (acting as a backup) [2,3]. Examples of meta-search engines are:

Ask Jeeves (<http://www.askjeeves.com/>),
Debriefing (<http://www.debriefing.com/>),
Dogpile (<http://www.dogpile.com/>),
Mamma (<http://www.mamma.com/>),
MetaCrawler (<http://www.go2net.com/search.html>), Verio
Metasearch (<http://search.verio.net/>),
Inference Find (<http://www.infind.com/>).

The construction of the Meta-search engine is simple. However, it is hard to improve its performance, for each search engine has different capacity with different search topic, area, etc. Its work state and information service quality changes continuously. Therefore, Meta-search engine should submit search work to the search engine according to dynamic situation.

2. THE MODEL OF INTELLIGENT META-SEARCH ENGINE

In the intelligent meta-search engine model, each independent search engine servers as a node of the information search service system. Its task is to perform users' information search inquisition. How to allocate the task to these search engines depends on analyzed historical record of each search engine.

Theoretically, search engine knowledge base containing complete information about each search engine is hard to realize. But we can resort to narrow the range of relative search engines, according to specific search topic. The knowledge base is a kind of approximate realization of the information search service. According to the certain topic, constructing a knowledge base is relatively easy. It cannot compare with ideal knowledge base on quantity and quality, but is practical for a specific topic-searching service. More importantly, the knowledge in the base can be added, improved and updated.

In the Intelligent meta-search engine system, it is divided into four parts, user interface manager, search engine manager, search agent and Data-Ming manager. Fig.1 illustrates the model of the system.

User Interface Manager (UIM)

User interface manager is responsible for receiving user inquisition. It interprets user operations with system's interface into standard systematic inquiry. After search engine manager integrates relative information, it will take the responsibility for exporting search results [4].

Search Engine Manager (SEM)

When search engine manager receives standard systematic inquiry. Sent by user interface manager, it will send search condition to Data-Ming manager and receive search engine agents schedule strategy. According to the dynamic strategy, connection requests are generated. Then each search engine agent can exert his power to seek relative information. At the same time, it provides with much more functions, such as buffer management and information integrity.

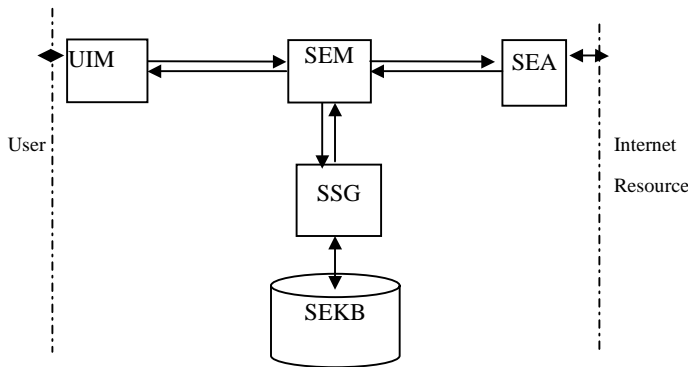


Fig.1 structure model of the intelligent meta-search engine system

Search Engine Agent (SEA)

There are several definitions of agents [5]. One can also describe rather than define agents in terms of their task, autonomy, and communication capabilities. Some of the major definitions and descriptions of agents are:

Agents are semi-autonomous computer programs that intelligently assist the user with computer applications. Agents employ artificial intelligence techniques to assist users with daily computer tasks, such as reading electronic mail, maintaining a calendar, and filing information. Agents learn through example-based reasoning and are able to improve their performance over time [6].

As a browser, each search engine agent is an independent process communicating with a search engine, which is the really part to get information resource on the Internet. So long as we send applications according to HTTP protocol to a Web server, the server will give corresponding reply.

Scheduling Strategies Generator (SSG)

Scheduling Strategies Generator is the main model of the Intelligent Meta-search Engine. It implements pre-process function and generates scheduling strategies with real-time condition. After having analyzed the technologies of data mining, decision tree for classification is chosen to generate scheduling strategies of search engines. In the next section, we will describe its process of generating scheduling strategies, the assessment Of Information Search Service and the realization With OLE DB.

3. DESIGN AND REALIZATION OF SCHEDULING STRATEGIES GENERATOR

Data mining has developed quickly in last few years. Techniques of nerve network, association rules, and clustering and decision tree has become major methods [7]. In this paper decision tree for classification is chosen to solve this problem. We can classify data (constructs a model) based on the training set and the values (class labels) in a classifying attribute and use it in classifying new data. Decision tree is divided into classification tree and regression tree. The former

deals with dispersed variables, and the latter copes with continuous variables.

Decision tree has a flow-chart-like tree structure. Its internal node denotes a test on an attribute and branch represents an outcome of the test. Leaf nodes represent class labels or class distribution. Tree construction is that all the training examples are at the root and partition examples recursively based on selected attributes. We need to identify or remove branches that reflect noise or outliers.

Using decision tree, we can classify an unknown sample with testing the attribute values of the sample against the decision tree.

Performance Evaluating Of Information Search Service

Data mining process will produce large quantity of information, among which only few are concerned to us. It is essential for us to build up criteria to judge the quality of information search service given by the independent search engines. That is to say, with technology of Data mining, we can judge whether a independent search engine is suitable to this search task or not by two term 'Precision' and 'Recall'.

There are three major information retrieval paradigms: statistical, semantic, and contextual. The first approach emphasizes statistical correlations of word counts in documents and document collections. The semantic approach to information retrieval views documents and queries as representing some underlying meaning [8, 9]. It emphasizes natural language processing or the use of artificial intelligence queries. The third approach takes advantage of the structural and contextual information typically available in retrieval systems.

Salton [10, 11] describes the use of statistical schemes such as vector space models for document representation and retrieval. The Smart system [12] is an example of a text processing and retrieval system based on the vector-processing model. Another example is Latent Semantic Indexing (LSI) [13], which captures the term associations in documents. Yunjae Jung describes an effective Term-Weighting scheme for information retrieval on the basis of the vector space model (VSM)[14,15].

Precision: Used to describe the relationship of the information searched by a search engine. Higher the precision value, the more accurate information search service can be offered by the search engine under this condition.

Recall: Used to describe the quantity of the information searched by a search engine. If a search engine has higher recall value with this search condition, it can offer plenty of information related to the content. Following is the algorithm.

In our proposed algorithm, the number of the independent search engines onto which a test information search task is given, is assumed to be unbounded. So we can add or delete a independent search engine from the target system dynamically. We temporally schedule m search engines with n records that are store in database table. The table can be modeled by a matrix $P(S,P)=(p_{ij})_{n \times m}$, where S is a set of s , representing the search engine, and P is a set of p , representing the record. Thus, p_{ij} represents the order of p_i searched by s_j . Then, determine the precedence constraints among the tasks. The weight value of p_i , denoted as $w(p_i)$, represents the value of

this information. The weight value of p_{ij} , denoted as $w(p_{ij})$, represents the value of this information searched by s_j . The maximum order of p_i is denoted as $Max(p_i)$. So we can conclude to following equations:

$$w(p_{ij}) = (1 - p_{ij} / \text{Max}(p_j)) \times P_{ij} \quad \dots (1)$$

$$w(p_i) = \sum_{j=1}^m w(p_{ij}) \quad \dots (2)$$

$$P(s_j) = \sum_{i=1}^n w(p_i) \times P_{ij} / n \quad \dots (3)$$

$$R(s_j) = \sum_{i=1}^n P_{ij} / n \quad \dots (4)$$

To summarize, the equations of Precision and Recall are given to assess the information search service of a independent search engine. The engine that has lower value will be considered as bad behavior, will not be chosen at next time with the same search condition.

The Process of Scheduling Strategies Generator

Decision tree generation consists of two phases. So Classification with decision tree is also a two-step process [16].

(1) Model Construction

In this step, we will describe a set of predetermined classes. For each sample is assumed to belong to a predefined class, as determined by the class label attribute. The set of tuples used for model construction is called as training set. The model is represented as classification rules, such as decision trees or mathematical formulae. So, we can use classification algorithms to analyze training data, which includes the precision and recall of search engines within certain condition, such as topic. Then, model of scheduling strategies can be made. Fig.2 illustrates the process of model construction.

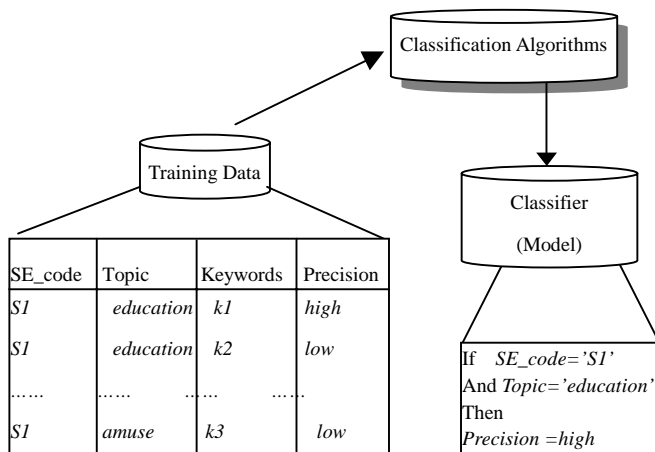


Fig.2 Process of Model Construction.

(2) Model Usage

In second step, we will use the constructed model to classify future or unknown objects. Firstly, we will estimate accuracy of the model. The known label of test sample is compared with the classified result from the model. Accuracy rate is the percentage of test set samples that are correctly classified by the model. Test set is independent of training set, otherwise over-fitting will occur. If the precision of unseen data is high, the search engine will be invoked to do the information search

task. Fig.3 illustrates the process of model usage.

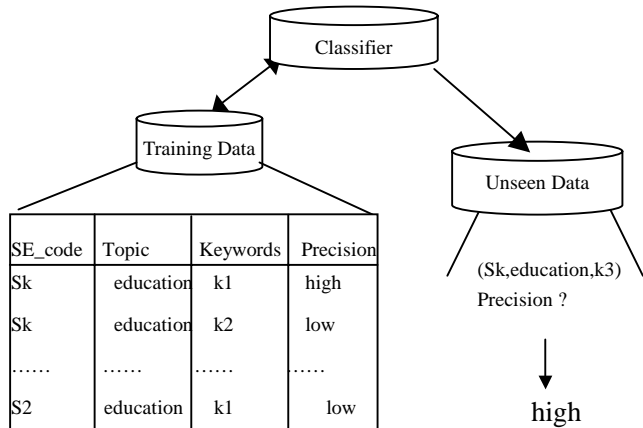


Fig.3 Process of Model Usage

Realization with OLE DB for DM

Microsoft cop. put a general language called as OLE DB for Data Mining (DM,) in March 2000[17]. It is capable of definition model of DM and communication with DM system as inquiry language. The specification of OLE DB for DM has offered a open interface for software business and developer. At the same time, it can combine with the PMML standard, which is issued by DMG cop. to take application of DM more powerful open standard. We also can find the support of decision tree algorithm in the Microsoft SQL SERVER 2000. Next, we will show how to generate scheduling strategy with OLE DB for DM [18].

(1) Create DMM Object with the Decision Tree Algorithm

Mining Model of DMM has been led into OLE DB for DM as a new fictitious object. It just likes a relation table in database, containing special rows. Trained data and predication rules will be stored in these rows. For instance, the DMM object named as SEPrediction is created in following steps.

```
Create Mining Model SEPrediction
(
  SEid long key,
  Language text discrete,
  area text discrete,
  topic text discrete,
  gain long continuous,
  ... ..
)
using microsoft_decision_trees
```

In the statements, *key* specifies the keyword *SEid*; *Discrete* shows that this property can be dispersed using the function of *DISCRETIZED()* that system offers to designate the specific dispersed function. *Continuous* shows the data to be continuously distributed. The keyword of *Using microsoft_decision_trees* means the decision tree algorithm of Microsoft should be selected.

(2) Packing Data to Model for Data Training

After creating the DMM model of *SEPrediction*, we can put data into it by using *Insert sentence*. Then, the decision tree algorithm, which we have selected, can start voluntarily to analyze the data and to generate the model of the decision tree

model of Intelligent Meta-search Engine.

(3) Using the Trained Model

After training, the model has the form of true value table. One or more lines of it correspond to a possible combination of each property row of DMM model.

When the inquisition has been changed to a standard search task, containing the information of the language, theme and area that has contained information search result, data are stored in *New Request* tables. *PREDICTION JOIN* sentence will join for searching information and training data. Then *SELECT* sentence can choose the corresponding information in model, to predicate Scheduling Strategy.

4. CONCLUSION AND RESEARCH PROSPECT

There is an increased amount of information available on the Web and an increase in the number of on-line databases. This information abundance increases the complexity of locating relevant information. The combination of the search and retrieval engines, the agent and the information retrieval algorithm addresses the trust and competence issues of intelligent meta-search engines. The user controlling the parameters and temporal existence of the search engines via the query of the search and retrieval engine ensures an element of trust. The user gets continual feedback from the system via the agent's prioritizing of the remaining query results, which addresses the competence issue.

A product information search subsystem is built up within a government bid management system with the intelligent search engine model to promote the performance of information search service. In the future, we will make study of how to control quantity of data for training set. Although it is proposed as a information search model, it also can be used to improve the quality of network Value-added information service [19].

5. REFERENCES

- [1]. Michael Krantz. Chiming in on Yahoo's roar, Mediaweek. 1998. vol. 6, no. 3, 9~12
- [2]. Barlow, Linda. The Spider's Apprentice: A Helpful Guide to Web Search Engines. <http://www.monash.com/spidap.html>. Sept. 2001.
- [3]. Elkordy, Angela. Web Searching, Sleuthing, and Sifting. http://www.thelearningsite.net/cyberlibrarian/searching/is_main.html. February 2000.
- [4]. Liu Li, Sun Yan-tang. Internet Information Search Service Based on the Technology of Data Mining. International Symposium on Distributed Computing and Applications to Business, Engineering and Science Proceedings. Hubei Science and Technology Press, 2001. 213~216.
- [5]. Marina Roesler and Donald T. Hawkins. Intelligent agents. 1994. Online, vol. 18, no. 4., 18~32.
- [6]. Linda Rosen. MIT Media Lab presents the interface agents symposium: Intelligent agents in your computer?. Information Today. 1993. vol. 10, no. 3, p. 10.
- [7]. Jiawei Han, et al. Concept and Technology of Data Mining. Beijing: Mechanical industrial press, 2001. 185~222.
- [8]. Gerald Salton, Amit Sanghal, Chris Buckley, and Mandar Mitra. Automatic Text Decomposition Using Text Segments and Text Themes. Hypertext 96. 1996. 53~65.
- [9]. Gerald Salton, Chris Buckley and Maria Smith. On the Application of Syntactic Methodologies in Automatic Text Analysis. Information Processing & Management. 2000. vol. 26, no. 1, 73~92.
- [10]. Gerald Salton, James Allan and Amit Singhal. Automatic text decomposition and structuring. Information Processing & Management. , 1996. vol. 32, no. 2, 127~138.
- [11]. Gerald Salton, James Allan and Chris Buckley. Automatic structuring and retrieval of large text files, Communications of the ACM. , 1994. vol. 37, no. 2, 97~108.
- [12]. Chris Buckley, James Allan and Gerald Salton. Automatic routing and retrieval using Smart: TREC-2. Information Processing & Management. 1995. vol. 31, no. 3, 315~326.
- [13]. Scott Deerwester, Susan T. Dumais, George W Furnas, Thomas K. Landauer and Richard Harshman. Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science. 1995. vol. 41, no. 6, 391~407.
- [14]. Y. JUNG, H. PARK, AND D. DU. An Effective Term-Weighting Scheme for Information Retrieval. Technical Report TR00-008. Department of Computer Science and Engineering, University of Minnesota.
- [15]. Y. Jung and D.Z. Du. A Balanced Term-Weighting Scheme For Improved Document Comparison And Classification. Proceedings for the first SIAM international workshop on text mining, Chicago, IL, April 7, 2001.
- [16]. H.D. Mittelmann and P. Spellucci. Decision Tree for Optimization Software. World Wide Web. <http://plato.asu.edu/guide.html>, 2004
- [17]. Surajit Chaudhuri, Usama Fayyad, Jeff Bernhardt. Scalable Classification over SQL Databases. ICDE 1999, 470~479.
- [18]. Microsoft, OLE DB for Data Mining Specifications. www.microsoft.com/data/oledb/dm. July 2000
- [19]. Liu Li, Xu Wen-bo. Technology Of Intelligent Meta-search Engine Applied In Network Information Value-added Service. International Symposium on Distributed Computing and Applications to Business, Engineering and Science Proceedings. Wuhan University of Technology Press, 2002. 256 ~ 259.

Li Liu, lecturer of College of Information Technology, Southern Yangtze University, BSc, computer network and application, Light Industry of Wuxi, China, 1999.7, research interests in network information system, artificial intelligence.

Wenbo Xu, Professor, tutor of Ph.D. students of Computer Engineering School of Information Technology, Southern Yangtze University, research interests in network information system, artificial intelligence and computer control system, the chairman of DCABES 2001

A Study of CORBA Multi Port ORB Architecture Based On Hierarchy Domain*

Guo Yinzhong, Xie Liping, Xu Yubin, Zeng Jianchao
Division of System Simulation and Computer Application,
Taiyuan Heavy Machinery Institute, Taiyuan, Shanxi, China. 030024
Email: guoyinzhong@263.net Tel: 0351-6220268

ABSTRACT

A configurable multi port ORB architecture based on hierarchy domain is proposed in this paper, after analyzing the ORB core of CORBA distributed computing platform and POA handling states semantics in CORBA standard. The composing of POA Domain, Event Handler Domain, CORBA Object Domain and the interaction among these domains are discussed at length in this architecture. At the same time the request handling states and their transformation in Event Handler are discussed. Users can deploy the distributed application and configure the distributed resources by domain.

Keywords: CORBA hierarchy domain multi port ORB architecture

1. INTRODUCTION

In current distributed network computing technology field, the distributed object computing technology, has become the main technology of solving the problems of objects interoperation, portability and platform-crossing integration in distributed heterogeneous platform. The technology is the production of combining the distributed computing technology with the object-oriented technology. Especially CORBA standard, which is based on distributed object technology and established by international Object Management Group (OMG), has become the main software standard in solving enterprise level distributed computing. In the development and design of the CORBA-based enterprise level distributed computing platform, distributed computing platform must have flexibly deployed architecture to adapt the diversity of distributed system, which can support user to divide Management domain of distributed system according to actual needs and configure distributed application and management distributed resources according to domain. In current CORBA standard, because of the opacity of ORB operation, ORB cannot support distributed application scheduling request broker according to application policies of system loading, resource limitation and task PRI, etc, and cannot provide the controlling request broker measures for user and dynamically deploy distributed resources system. Therefore, aiming at the limitation of traditional ORB architecture in distributed application deployment form, this paper proposes a hierarchy domain-based multi port ORB architecture, discusses the composing of POA Domain, Event Handler Domain, CORBA Object Domain and the interaction among these domains, and studies the request

handling states and states transformation in Event Handler Domain. That is researched in order to solve the conflict between user management requirement and distributed application development form in distributed system.

2. TRADITIONAL ORB ARCHITECTURE AND ITS PROBLEMS

CORBA is an implementation standard of Object Management Architecture (OMA) defined by Object Management Group (OMG), which is an open distributed computing software standard based on distributed object technology. Figure 1 shows its architecture.

When client needs to request object service, he can use client IDL stubs or Dynamic Invocation Interface (DII) to activate remote object's local broker, and at the same time the message of object request broker is constructed, which is transferred to server side by ORB Interface though ORB soft bus. When detected the network event, ORB of the server side sets up network connection and generates the corresponding processing to deal with the events of the connection. Object request broker (ORB) is the core of CORBA architecture. ORB core manages the bottom communication mechanism, dispenses IIOP message, helps Object Adaptor (OA) handle request. And it wraps IIOP engine, codes and decodes messages according to CDR (Common Data Representation) rule. ORB Interface separates application from ORB implement details, and provides ORB common services to application. Traditional ORB architecture, which is architecture of singly deploying distributed resources and providing single interface to user, does not support users to divide domains according to policies and configure distributed application according to their actual need. All object references generated by ORB have the same listener port, processing the events with ORB core is not controlled by POA. This kind of ORB architecture has the following problems:

- 1) Single environment deployment and single port configuration in ORB architecture does not make distributed application manage distributed resources according to application policies, and is not satisfied with the needs of distributed application.
- 2) When request arrives the relevant POA request queue in ORB architecture, ORB core has to unmarshal the request header and adapt the POA. When the queue is full or request is in the discarded state, the request will be discarded, original process is cancelled, which wastes system resource.
- 3) The Implementation of ORB flow control mechanism needs maintain a lot of information and complex controls. Even in single thread, POA has to maintain request source information in request queue, and return the result to user by complex control.

* This paper is supported by Shanxi young Science fund (ID: 20031029)

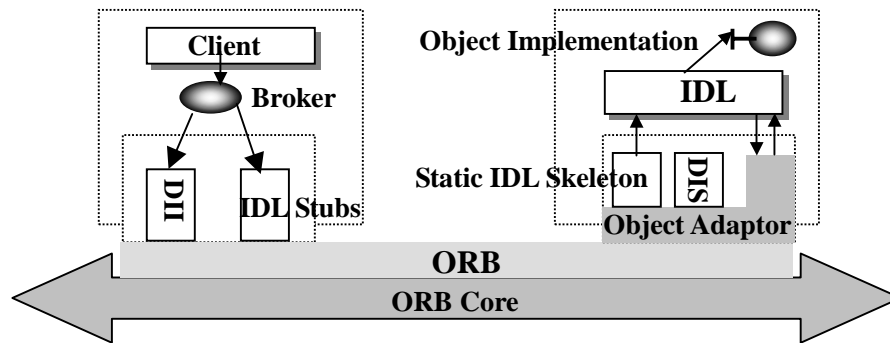


Fig. 1. CORBA Architecture

In order to solve these problems above, new research on ORB architecture is required in the development of distributed computing platform. The paper proposed a hierarchy domain-based multi port ORB architecture, which is based on the analysis of the concept model and the relationship model of ORB role and POA handling states semantic definition in CORBA standard.

3. HIBERARCHY DOMAIN-BASED CORBA MULTI PORT ORB ARCHITECTURE

3.1 ORB core Running Mechanism

ORB core primarily manages the communication mechanism of bottom level, and takes charge of coding and decoding messages. ORB separates the activation of client object request from object implementation, mapping object request to the method of this request transparently. This kind of network-based request mapping of ORB core can be formally described as:

ORB Core = < Events, states, rules, roles >

ORB event set: Events= {network connection event, network read event, network write event, network fail event, interaction event}

ORB core states: States= {listener port state, network connection state, human-machine interaction state, ...}

ORB event handling rules:
Rules={ $Events \times States \rightarrow Rules$ } event types and ORB core state determine the rule of the handling event.

ORB core roles: Roles={event reactor (Reactor), connection request handler (AcceptorHandler), network read/write handler (RWHandler), human-machine interaction event handler (MHandler)}

ORB core running mechanism is as follow: event reactor (Reactor) in ORB core react various events in ORB event set through Select mechanism or windows message mechanism supported by system, then the various events are mapped to the corresponding event handler through event handling rules (Rules) to handle the related events. For network connection event, AcceptorHandler sets up the connection, and creates the Wrangler to handle the events of the connection; For network reading event, RWHandler has the reading operation on the related connection, and gives the

request to the relevant adaptor to hand it out; For network writing event, RWHandler reads the message from the related output buffer, executes the writing event, and revise the Rules; For network invalidation event, RWHandler closes network connection, revises States and Rules. Handling all kinds of events are implemented by network communication mechanism of system TCP/IP protocol.

3.2 Related concept definition

The following defines some concept about hierarchy domain-based multi port ORB architecture.

Definition 1: POA domain: marked as *POADomain*, a set of all POAs managed by a POAManager. Object adaptor level may contain multi *POADomain*. $POADomain_i$, $POADomain_j =$.

Definition 2: Event Handler domain: marked as *Eventhandler Domain*, corresponding to the *POADomain* of OA level, has its own listener port and contains AcceptorHandler of the port and a set of RWHandlers of the port connection.

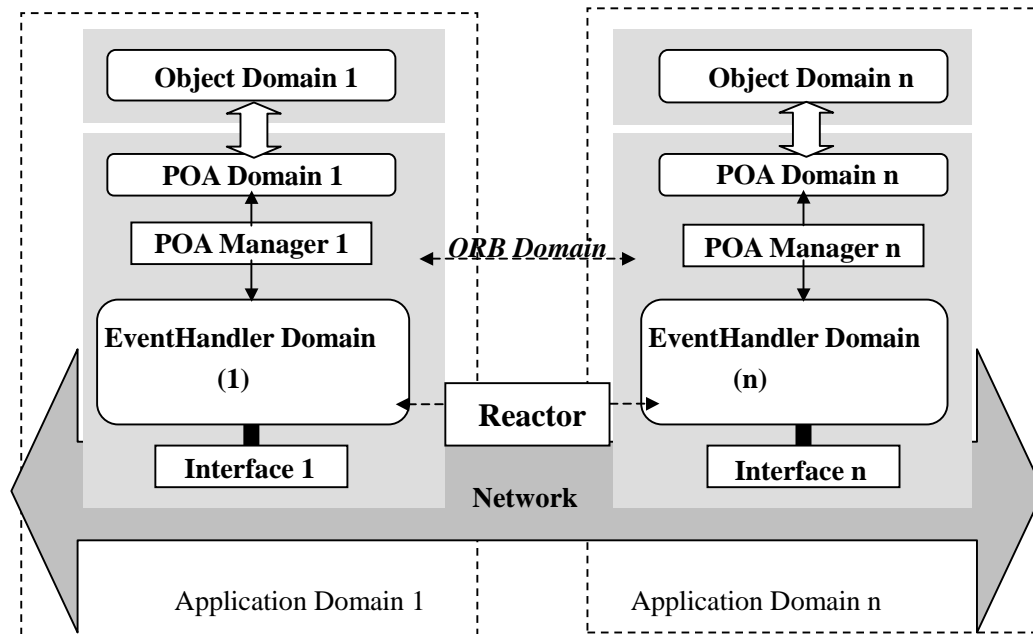
Definition 3: ORB domain: marked as *ORB Domain*, composed of *POADomain* and the relevant EventHandler domain, between which object request mappings are completed independently.

Definition 4: CORBA object domain: marked as *Object Domain*, Object Domain={Object | Register(object) POA Domain}, Object represents CORBA object, Register function return the POA of the generated CORBA object.

Definition 5: Distributed application domain: marked as *Application Domain*, consists of ORB Domain and the related CORBA Object Domain, which is divided by a certain policy and application need of the user.

3.3 Domain-based multi port ORB architecture

In distributed application system, domain-based multi port ORB architecture in figure2 is proposed. The architecture supports user to divide distributed application domains according to a certain policy and application need and deploys distributed resource in different level domain. Each distributed application domain consists of *POADomain*, EventHandler Domain and Object Domain, all of which



implement their division policies. In **POADomain**, each POA has the same request handling state. In corresponding **Object Domain**, POAManager controls the related CORBA object request handling. **POAManager** manages the related **POADomain**, and AcceptorHandler of the related **EventHandlerDomain**. Each **POADomain** in the OA level corresponds to an **EventHandlerDomain**, which has its own listener port. Each **EventHandlerDomain**, invoked by **EventReactor**, independently provides services to upper domains. Different application domains have different ports; user can transparently use domains or explicitly deploy distributed applications by domains according to different ports. The domain-based hierarchy architecture, simplifies ORB inner part implementation and interaction control among levels, avoids complex controlling between ORB core and adaptor levels and implementing flow control mechanism efficiently.

In domain based hierarchy multi port ORB architecture, each ORB management and application domain has the same handling state. Each RWHandler in *EventHandlerDomain* maintains a request queue; POAManager manages the AcceptorHandler in its corresponding *EventHandlerDomain*. When POAManager is initialized, an AcceptorHandler is created in a new port, which is managed by the POAManager. When a new domain is created, AcceptorHandler controls and manages a pair of RWHandler corresponding to the port. When POAManager handling state has been changed, the corresponding AcceptorHandler and RWHandler handling state are changed. In the hierarchy architecture, each *POADomain*, controlled by POAManager, transfers and controls handling states.

In application domain, each **EventHandlerDomain** corresponds with POADomain; they have the same request handling states and states transformation. Each ORBDomain and application Domain has the same handling state. In

EventHandlerDomain, there are four states of handling request: Active, Inactive, Holding and Discarding. In different request state, the states of Acceptorhandler and Wrangler are different. In Active state, Acceptorhandler reacts connection request, creates an Acceptorhandler corresponding to this state. Acceptorhandler completes network reading and writing events, adds request message to the queue, pops the request from the queue, decodes the request header, and gives the request to OA level to handle. In Inactive state, Acceptorhandler refuses connection request and notifies all of RWHandlers to do deactivate operation, which operation is closing the connection. RWHandler gives out all response messages, including Close Connection message, and closes the connection. For holding and discarding states, Acceptorhandler responses connection request, creates the corresponding state of a RWHandler. CORBA multi port ORB architecture based on Hierarchy domain makes up for the weakness of the original ORB architecture and provides user a flexibly configurable multi port service. Each POADomain has its independent port, CORBA objects in different application domain have the different binding ports, which makes user easily deploy and configure distributed application according to domains. Therefore unhandled request queue is no longer handled, system resources utilization is improved, request source information needn't to be maintained, and architecture control is simple.

4. CONCLUSION

During the design and research of distributed computing platform, hierarchy domain-based multi port ORB architecture is proposed in this paper, in virtue of the diversity of distributed system application needs. The architecture supports user to divide distributed system application domains according to a certain policy and

application needs, to deploy distributed applications and to configure distributed resources. Multi port ORB domain architecture mends the disadvantages in the original ORB architecture, such as low utilization of system resource, complexity of control among different levels, single deployment of single port. The architecture makes user deploy and configure distributed application transparently according to domain and provides user a method of controlling request handling and a mechanism of responding dynamic changes of resource.

5. REFERENCES

- [1]. Object Manager Group Common Object Request Broker Architecture and Specification Version 2.41 2002
- [2]. Aniruddha Gokhale Irfan Pyarali Carlos O`Ryan ET. Design Considerations and Performance Optimizations for Realtime ORBs in: COOTS`99 San Diego CA.1999
- [3]. Xiu-Chuang Wu Jin-Bin Ju Interfacing WPRS to the Grid System in: ICA3PP2002 Beijing Oct.22-25.2002
- [4]. Xiang Jun Zhong Li ET. Research on Stream Control in CORBA Platforms Based On React We Agent Model Journal of Computer Research & Development (in Chinese), 2001, 38(2)
- [5]. K.Dincer and G Cfox, Design Issues in Building Web Based Parallel Programming Environments Six International Symposium on High Performance Distributed Computing August 1997
- [6]. Yin Zhang Guo Hai Yun Fang ET. Research of Object Binding and Request Invocation Mechanism of CORBA Distributed System In: ICA3PP2002 Beijing Oct.22-25.2002



Guo Yinzhong, male, birth in 1969, adjunct professor and major study: computer cross-network technique and distributed network computing.

Xie Liping, female, graduate student.

A Multi-purpose Web Information Publishing Framework

Yiqing KONG, Bin FENG, Wenbo XU, Shitong WANG
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, 214036, China

ABSTRACT

For the reason that more and more materials of various kinds have and will be published on the web, it is quite natural for us to build up a web information-publishing framework, which responds with a published version of files according to a specific request. Because of not only the frequent changes as well as different file formats, but also the demanding of dynamic pages, the making of a multi-purpose framework is needed, which will fulfill the task of extracting information from various kinds of web pages, such as HTML files, and XML files, also from databases, then using certain technologies to generate output files, formatting ranging from XML, HTML, text, VRML, WML, VoxML and PDF, to those formats to be produced in the near future.

Keywords: Information, Publishing, Multi-purpose Framework, Cocoon.

1. INTRODUCTION

The rapid development of Internet has successfully changed the representation of almost all information from paper-based style to web-based electronic format. In the following several years, virtually nearly every major application will be completely web-based. At the same time, users are demanding more functionality while marketing are pursuing more flexibility in looking and feeling of representation.

The frequent changes of information to be represented on the web have made a vast request that a complete application or the design of a web site renew as often as once a week, usually forcing the web designers to spend days changing hundreds of web pages, therefore, asking computer developers to depend on the technology of dynamic information publishing.

Now more and more people are using XML technology to develop web pages. XML is usually referred to as portable data in the sense that XML parsing is application-independent.

The same XML parser will read every possible XML document: one describing a bank account, another representing an Italian meal, etc. This is quite impossible with other text-based or binary file formats, in other words, XML is much more flexible and structured for the use of knowledge representation whether for structured databases or for applications based-on web.

Because of its advantages of flexibility and structure, XML is a much more suitable language for information publishing than any other languages. However, having hundreds of XML documents on a web site does no good if there were no mechanism to apply transformations on them when requested. To output XML documents that should be consistently styled; a web information publishing framework will on some degree improve the situation.

2. ISSUES TO BE CONCERNED

Considering the ever-changing information applications demanding, the web information publishing framework should address those complicated issues ranging from allowing to toss out long hours of HTML coding, by using XML documents to replace HTML, styled by XSL documents, to those converting all complicated web designing works into a piece of cake, and allowing applications to be able to change looking and feeling as often as want.

Just as a web server is responsible for responding to an URL request for a file, the web publishing framework is responsible for responding to a similar request; however, instead of responding with a file, it will often respond with a published version of files. In such cases, a published file refers to a file that may have been transformed with XSLT, or converted into another format such as a PDF file. The requestor does not have the request of seeing the raw data that may underlie the being published results.

3. BROWSER-DEPENDENT STYLING

In addition to specifically requesting certain types of transformations, such as a conversion to a PDF file, the framework should allow for dynamic processing to occur depending on the request, applying different formatting based on the media of the client. In the web environment, this would allow an XML document to be transformed differently based on the browser being used. A client using Internet Explorer could be served a different representation from a client using Netscape; or different styles among versions of HTML, DHTML, JavaScript, and other such things. For the wide range of browsers, this feature will be available in the web publishing framework, so that it could achieve the aim of browser-dependence.

Besides providing built-in support for many common browser types, the framework is in want of compatibility for other browser types newly developed and to be developed in the future.

With the recent extending of wireless devices, one of the real powers in the dynamic application of style sheets lies in the possibility of using of these devices. As wireless devices make use of WAP and WML pages for their browsing, the output formatting possibilities by no means should not exclude WML as well.

4. BLOCKS OF WEB PAGES

In the traditional web presentation technology, web pages are under widely usage. Web pages include almost every thing, from the truly demanded data to only-for-display styles, hence, making the trouble of rather difficulty of information retrieval.

However, it is quite fortunate that, that is not such a headache. As a matter of fact, web pages could be divided into several parts, with each part in charging of one specific task, therefore, we could tell that, there exist blocks in web pages.

Web pages could be separated into three parts: that is, Layout, Content and Logic. But in most cases, layout, content and logic are merged together.

The problem is that, usually, layout, content and logic of web pages of sites change frequently and independently. For example, thought of web pages with colors, logos, header, footer, e-mail addresses, business model, shopping cart implementation, and bonus programs; layout, content and logic are merged together in these pages, producing a rather rough work to try to make changes to these pages. Considering a large web site with more than 200 pages, it is an absolute difficulty to change every page; therefore, maintainability and flexibility are of great importance in these web sites.

The solution to such troubles is to make use of separation of layout, content and logic.

Separation of such concerns is that web designers are in charge of layout, while domain experts are in consideration of content and computer programmers are in position of logic. Content will be changed as frequently as possible; in the meantime, layout and logic will be achieved the goal of how to get reused of. There is no denying the fact that it is a very method of solving the difficult problem of maintenance and flexibility of web-based applications.

5. CAPITALIZE ON COCOON

Cocoon [1] is in charge of XML-based web publishing, which divides responsibilities for logic, presentation and management, facilitating web sites' creation, management, transformation and delivery of XML sources. The Cocoon publishing model is based on XSLT that provides a complete separation of the representation of content and style from the business logic.

To achieve such goals, the Cocoon model divides the development of web contents into three separate levels:

1. XML creation

The XML files are created by the content owners, but they do not require specific knowledge on how the XML contents are further processed rather than the particular chosen DTD/namespace.

2. XML processing

The requested XML files are processed and the logic contained in their logic sheet is applied. Unlike other dynamic content generators, the logic is separated from the content files.

3. XSL rendering

Then the created documents are rendered by applying an XSL style sheet to them and formatting them to the specified resource types (HTML, PDF, XML, WML, XHTML).

Cocoon has used to create HTML from XML derived from static sources or dynamically formatting, such as XSL: FO rendering to PDF, client-depending transformations, such as

WML formatting for WAP-enabled devices or direct XML serving to XML and XSL compatible clients. Server-side applications can deliver contents to any browser from XML sources.

XSL: FO is an XML DTD for describing 2D layout of text in both printed and digital media, which is a transformation used to generate a formatting object description of a document starting from a general XML file.

From XML input, the output formatting possibilities include: HTML, XML, and text, VRML, WML, VoxML and XSL-FO rendered into PDF. For more sophisticated XSL-FO tasks, formatting objects and graphics specifications, such as SVG, can be mixed and rendered into PDF. One XML document can be transformed to either HTML or to WML, if the page request came from a WAP-browser on a wireless device.

The use of XSP (eXtensible Server Pages) pages will provide a way for developing more sophisticated solutions for generating dynamic contents. XSP is a technology for building web applications based on dynamic XML content, being integrated into the Apache Web server for building web applications, allowing XML developers and designers to rapidly develop and easily maintain dynamic web pages that leverage XML applications. The XSP technology connects underlying databases and application modules using the Java programming language. The application logic resides in middleware resources, e.g. database access, database search and the combination of search results is coded in Java and executed in a middle-tier java virtual machine.

XSP pages are XML documents containing processing tags that instruct the applications to build dynamic contents for the users' request. An XSP page can build an XML document dynamically from multiple sources. The resulting documents will undergo XSLT transformations and deliver to the web browser. XSP built-in processing tags or user-defined library tags can be used to embed procedural logic, substitute expressions and dynamically build XML nodes. User-defined library tags act as templates that determine what program codes are generated from information encoded in the dynamic tags.

Dynamic web content generation is fully supported by XSP. Consider the following example:

```
<p>
  Good
  <xsp:logic>
    String timeOfDay = (new
SimpleDateFormat("aa")).format(new Date());
    if (timeOfDay.equals("AM")) {
      <xsp:content>Morning</xsp:content>
    } else {
      <xsp:content>Afternoon</xsp:content>
    }
  </xsp:logic>
</p>
```

By XSP processing, this XML fragment will before noon, generate:

```
<p>Good Morning!</p>
```

and after noon, generate:

```
<p>Good Afternoon!</p>
```

Besides XSP, the use of FOP (Formatting Objects Processor) is a XSL formatting tool for server-side processing; FOP is a Java print formatter driven by XSL formatting objects. FOP reads a formatting object tree and then creates a PDF document. From this technique, the system can support a variety of output formats.

6. THE MULTI-PURPOSE FRAMEWORK

6.1 Transformation from HTML to XML

Why Should Make a Change

Cocoon is in charge of XML-based web publishing, facilitating web sites' creation, management, transformation and delivery of XML sources. While for most resources created and to be developed for representation on the web, they are HTML pages, only suitable for applications of semi-structured data. For more and more people using XML technology to develop web pages, XML is usually referred to much more flexible and structured for the use of knowledge representation whether for structured databases or for applications based-on web, therefore, when considering of both semi-structured and structured data, it is better to use XML for data representation, demanding transformation from HTML pages to XML data.

To achieve this goal, a middleware is needed, being served as a wrapper [2], fulfilling the function of transforming different resources of HTML pages to easy-use, flexible and structured XML files by parsing data in HTML pages, changing into tree-structured DOM, and thus extracting information. The information represented by using HTML would be converted into XML data.

How to Achieve the Goal of Change

For HTML pages' semi-structured and something of tree-structured, information could be extracted from them [3]. Considering a HTML page as follows:

```
<HTML>...
<BODY>...
<IMG>
<H1>...
<TABLE>
<TR>
<TD>1</TD>
<TD>2</TD>
...
<TD>3</TD>
<TD>4</TD>
```

The structure in this HTML file can be demonstrated in figure 1.



Figure 1 The structure in this HTML

Suppose using “.” to demonstrate the inheritance of nodes, while “->” to demonstrate the data flow. For instance,
 html.body.img[0].getAttr(src)
 html.body.table[0].tr[1].td[0].3[0].txt
 The same node could be reached by different ways:
 html->img[0].getAttr(src)
 html.h1[0]->img[0].getAttr(src)

Integrated to the Framework

By capitalizing on Cocoon, the output formatting possibilities could include HTML, XML, text, VRML, WML, VoxML and PDF. However, for the input formatting, both HTML and XML would be possible. To make the multi-purpose framework useful, wrapper for turning HTML files into XML files is to be integrated [4].

6.2 The Architecture of the Framework

As all the issues concerned before, a multi-purpose framework will fulfill the task of extracting information from HTML files, XML files and databases, using XML Parser, XSLT and XSP technology to produce output files, formatting ranging from XML, HTML, text, VRML, WML, VoxML and PDF, to those to be produced in the near future.

The system architecture of the framework could be simply demonstrated in figure 2.

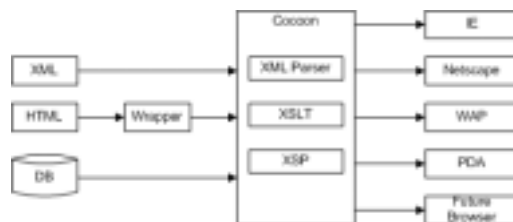


Figure 2 The system architecture of the framework

7. CONCLUSIONS

The application of a multi-purpose web information publishing framework will accomplish the task of publishing materials of various kinds on the web. This framework will extract information from different web pages, such as HTML files, XML files, also from databases, and use certain technologies to produce output files. The formats of these output files include not only all those existed, but also those to be created. Then it will on some degree achieve more functionality for advanced usage, together with more flexibility in information representation.

8. REFERENCES

- [1] <http://xml.apache.org>
- [2] <http://www.cs.washington.edu/homes/weld/wrappers.html>
- [3] <http://www.tropea-inc.com/technology/e4f/>
- [4] Meng Xiaofeng, "An Overview of Web Data Management", Journal of Computer Research and Development, Vol.38, No.4, April 2001, pp.385~395.

A Mediation-based Approach for Distributed Digital Library Services

Yu Jianghong¹, Fang Wei²

¹Zhuzhou Institute of technology, Hunan, China, 412008

E-mail: zz_yjh@163.com

²Wuhan University of Technology, Hubei, China, 430070

E-mail:80angel@163.com

ABSTRACT

In a distributed library environment, we can distinguish between metadata for information objects (e.g., documents) and metadata for information services (e.g., search services). In this paper, we have developed a working prototype library system. Digital libraries system (DLS) was developed and distributed as part of the Zhuzhou Institute of Technology's Electronic Libraries Programme and supports the rapid development of subject-based gateways. The system also provides a description of an approach for supporting the user or personalization process in digital catalogue systems.

Keywords: distributed protocol; Z39.50, Digital library system (DLS)

1. INTRODUCTION

As traditional libraries have evolved to meet the needs of their patrons, librarians have encountered and addressed a host of metadata-related issues. Today, sophisticated library cataloging principles and schemes help all of us in finding the information that we need in our local library. As work on digital libraries progresses, however, new metadata needs are arising. The Digital Library is a freely accessible facility that makes available a dozen or so quasi-independent collections of information over the World Wide Web many of which involve several Gbytes of on-line information [8]. The collections are intended as independent demonstrations of digital library technology, rather than a unified library in their own right. The integration of digital libraries (DL), data grids, and persistent archives is actively underway. While each community focuses on a different aspect of data management (data publication, data sharing, and data preservation) common software infrastructure is emerging.

Use of the open source Greenstone Digital Library software is gathering pace. By mid-2000, more than a dozen libraries and universities had arranged access to the software to help meet their digital library needs [5].

As a digital library grows and the variety of different kinds of material in it expands, problems of administration and maintenance become increasingly severe. As the user base expands, the collective needs of users expand too. To step outside the mind set indoctrinated by: generate a new Web page in response to a user clicking on a button or hyperlink—the classical form for a digital library, if you will—Greenstone, like other digital library projects [1, 2], provides a protocol for fine-grained interaction with other programs [3]. The protocol is implemented using CORBA [4] and has been extended to support both the SDLIP and Z39.50 protocols.

In this paper, we have developed a working prototype library

system. Digital libraries system (DLS) was developed and distributed as part of the Zhuzhou Institute of Technology's Electronic Libraries Programmer and supports the rapid development of subject-based gateways. The purpose of this paper is to demonstrate how distributed protocols promote a variety of distributed digital library applications.

The next section will give a brief overview of the related work. Section 3 introduces a system architecture. Section 4 we present a digital library for one's homepage and show the a strategy for supporting customization. Finally, we draw some conclusions in section 5.

2. RELATED WORK

In a distributed library environment, we can distinguish between metadata for information objects (e.g., documents) and metadata for information services (e.g., search services). The encoding of metadata for information objects can facilitate the unification of heterogeneous information objects, while the encoding of metadata for information services can facilitate communication among disparate services.

Many recent reports have concentrated on the metadata for information objects. Work on this aspect of the metadata problem includes the Warwick Framework [11], and the Jet Propulsion Laboratory's DARE metadata model [12].

Interoperability for digital library services was extensively addressed in the Stanford InfoBus [5]. In Info Bus, every service like search, attribute translation or metadata service is a distributed object implementing a well-known interface. In contrast, our framework focuses on a mediation-based approach. Furthermore, wrapper implementers do not have to agree on a set of common application-specific interfaces. The extensibility of the infrastructure is increased by relaxing the rigid low-level interfaces typical in distributed computing. A detailed comparison of approaches to interoperability for digital libraries can be found in [6].

Digital libraries inherently involve long distances between clients and collections. Concerns about response time, document availability, and network load in such an environment raise the question of whether stateful or stateless server operation is most appropriate.

In a stateful system, clients establish a session and then perform searches and explore the result set within that session context. The server promises to maintain the result sets until the end of the session. Before the advent of HTTP, many information access protocols (for example, Z39.50) were stateful.

Our architecture borrows from several areas of computer science. We can thus draw on a variety of traditions and

experiences. Component based software engineering is one such tradition [7]. Some commercial databases use data blades, analogous to our mediator blades. Experimental database designs have gone beyond these more limited blade facilities to introduce far reaching extensibility.

Our approach is heavily based on interface description. DL systems also emphasize the separation of interface and implementation.

3. SYSTEM ARCHITECTURE

The structure of the digital library is shown in fig.1.

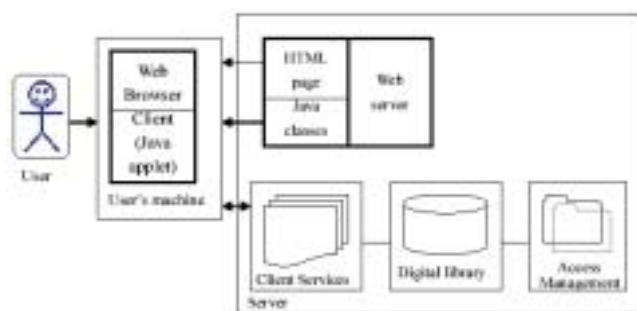


Fig.1. The Structure of the Digital Library System

We use the following Distributed System Topology:

- Use a web-based intranet in the field to catalog status. Employ Java-script to verify client-side info while providing server-side scripting via VBScript. Receiving station finds specialists through an LDAP server.

- Use JavaBeans on the client that communication with CORBA-enabled server-side trading services.

- Provide field with hand-held devices that collect multimedia data and transmit across an X.500 protocol to the doctors, bypassing indirection delays.

- Receiving-side doctors use ASP-generated pages to produce diagnosis from write-once, run-anywhere applets running on mobile devices on the field side.

- Mobile devices running Windows CE with ActiveX components talking to Perl-enabled Apache serves to provide asynchronous data feed. Fault tolerance and security provided by Kerberos integrated via a Python library on the backend.

- Prioritized TCP/IP packets over IPv6 addressing to ensure fast delivery in a peer-peer environment running on protected ports.

The DL is based on a few components. Inter-component communication is network based, so each one can be placed on a different host, however we do not use any middle ware. All communication is based on standard or proprietary protocols. The main component of this system is the Oracle database which stores all metadata. This describes the library content and user privileges. DL logic was implemented using a server stored procedures written in PL/SQL. DL logic means rules imposed on library and document structure, user privileges and searching capabilities. The last of these servers

has an application for the presentation and searching of library documents through the web interface. The application allows the loading of new documents. The document can be assembled by putting together objects of different types (text, images, video, audio) and defining the structure of the document manually dividing it into chapters and pages containing selected objects. The application uses SQL*Net protocol and JDBC driver for communication with database.

This system responds to all of the requirements specified earlier and allows the realization of services it was developed for distributed library and interoperability. Users can see all of the content through one entry point. In the future, interoperability with other libraries should also be provided Quality of Service. New methods for the presentation of multimedia objects according to a user's requirements and technical possibilities need to be developed. This is especially true for access network bandwidth and streaming media.

Performance and resource management. For mass usage of new services we need a new model of performance and content servers resource management. We also have to develop content caching and network multicasting to improve network usage access control. We also have to develop new methods for access control.

4. DIGITAL LIBRARY FOR ONE' HOMEPAGE

Figure 2 shows the home page of the Yuan's Digital Library, part of our University's digital library project. The vertical column in the center gives the collections available to the user. On the left are support services: a workspace for creative writing; a submission process for completed stories and poems; a bulletin board where selected works are discussed and annotated; and on-line training packages to help users learn about the digital library environment.

The receptionist asks the user to log in before reaching this page; in this case the user is Yuan, shown at the top of the page. There is also a special account for the class teacher, with extra functionality provided by the receptionist for updating collections with new stories and so forth. Authentication is not part of the protocol; instead it is built into the receptionist's software architecture.



Fig.2. The Zhuzhou Institute of Technology Yuan's Digital Library Environment

5. A STRATEGY FOR SUPPORTING CUSTOMIZATION

This approach for supporting customization and personalization in digital catalogues is illustrated in Figure 3. This system provides a description of an approach for supporting the user or personalization process in digital catalogue systems. We envision three primary customization operations as being important to support customization within digital catalogue systems. Firstly, a book value can be specified to replace an existing descriptive metadata field of a digital catalogue record.

Secondly, a user might want to delete or hide a metadata field of a descriptive catalogue record if it is not relevant for their purpose or task at hand. The third way in which digital catalogue information can be personalized is to define a completely new metadata field for a digital catalogue record.

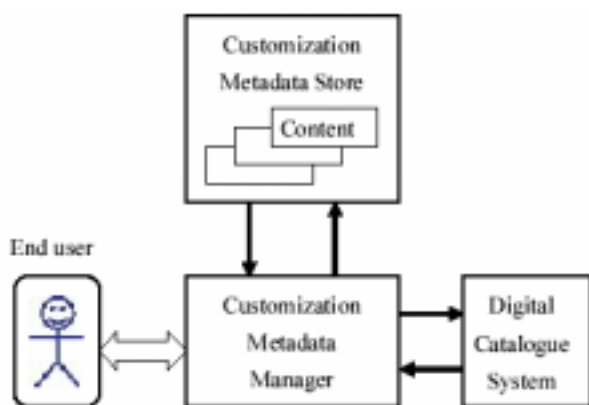


Figure.3. Supporting Customization of Digital Catalogue Information

6. CONCLUSIONS

We have developed a working prototype library system. Digital libraries system was (DLS) developed and distributed as part of the Zhuzhou Institute of Technology's Electronic Libraries Programmer and supports the rapid development of subject-based gateways. We are using it as a rapid prototyping environment for developing and testing the library's technical and user-centered design issues. This gateway is currently being used to support our metadata design and prototype tested cataloging efforts.

The main component of this system is the Oracle database which stores all metadata. This describes the library content and user privileges. DL logic was implemented using a server stored procedures written in PL/SQL. DL logic means rules imposed on library and document structure, user privileges and searching capabilities. This system responds to all of the requirements specified earlier and allows the realization of services it was developed for.

7 . REFERENCES

[1]. C. Logoze and D. Fielding. Defining collections in distributed digital libraries. D-Lib Magazine, 4(11), Nov.

1998. Available on-line at www.dlib.org/dlib/november98/lagoze/11lagoze.html.

[2]. A. Paepcke, R. Brandri_, G. Janee, R. Larson, B. Ludaescher, S. Melnik, and S. Raghavan. Search middleware and the simple digital library interoperability protocol. D-Lib Magazine, 6(3), Mar. 2000. Available on-line at www.dlib.org/dlib/march00/paepcke/03paepcke.html.

[3]. R. McNab, I. Witten, and S. Boddie. A distributed digital library architecture incorporating different index styles. In Proc. IEEE International Forum on Research and Technology Advances in Digital Libraries, pages 36–45, Santa Barbara, California, 1998. IEEE Computer Society Press.

[4]. D. Slama, J. Garbis, and P. Russell. Enterprise CORBA. Prentice Hall, 1999.

[5]. A. Paepcke, M. Baldonado, C. Chang, S. Cousins, and H. Garcia-Molina. Using Distributed Objects to Build the Stanford Digital Library Infobus. IEEE Computer, February 1999.

[6]. A. Paepcke, C.-C. K. Chang, H. Garcia-Molina, and T. Winograd. Interoperability for digital libraries worldwide. Communications of the ACM, 41(4): , April 1998.

[7]. E. Gamma, R. Helm, R. Johnson, J. Vissides, and G. Booch. Design Patterns. Addison-Wesley, 1995.

[8]. Witten, I.H., Cunningham, S.J. and Apperley, M.D. (1996) "The New Zealand Digital Library project." New Zealand Libraries 4 8(8), 146–152.

[9]. National Information Standards Organization (1995) Information retrieval (Z39.50): application service definition and protocol specification. ANSI/NISO, Bethesda, Md.

[10]. The OMG Homepage. <http://www.omg.org>, 1999.

[11]. Carl Lagoze, Clifford A. Lynch, and Ron Daniel, Jr. The Warwick Framework: A container architecture for aggregating sets of metadata. Technical Report TR96-1593, Cornell University, Computer Science Dept., June 1996

Web-Based Learning and Fault Diagnostic System

Feng Pan and Wenbo Xu

School of Information & Control Engineering, Southern Yangtze University

Wuxi, Jiangsu, 214036, China

Email: panfengx@pub.wx.jsinfo.net Tel.: 0510-8883618

ABSTRACT

This paper presents a novel knowledge-based multi-agent system for remote fault diagnosis, which is composed of learning and diagnostic agents (LDAs), machine agents (MAs) and a central management agent (CMA). Machines are remotely diagnosed by the LDAs through the communication channels between the MAs and the LDAs. When faults that cannot be solved with the present knowledge base occur, the DLA can acquire new knowledge, translate it into rules using a rule builder, and update the rules into the CKB. The CKB will become mature through a continuous learning process. A prototype system has been developed and used for remote fault diagnostics of tool wear in computer numerically controlled (CNC) machining.

Keywords: Expert systems; Fault diagnosis; Knowledge acquisition; Multi-agent systems

1. INTRODUCTION

Manufacturing industries are facing serious structural problems brought about by rapid developments of overseas activities and manufacturing factories. Factories located in different regions must be coordinated through the use of state-of-the-art information technologies to ensure consistent product quality. As a result, manufacturing activities can be integrated and monitored in many regions and countries. The performance of a machine could be monitored and accessed from anywhere in the world. In addition, information on productivity, diagnostics, and training of manufacturing systems could be shared among partners at different locations. The development of a remote diagnostics system would provide manufacturers and users with greater flexibility for conducting manufacturing activities.

At present, remote diagnosis through the Internet does not have many practical applications in shop-floor manufacturing. In addition, many of the knowledge bases used in the diagnostic systems are traditional rule bases. That is, before a knowledge base is put into use, it is built to be large enough to include as much knowledge as possible. During the working process, new knowledge that may be added to one knowledge base would still be unknown to the others. If there is a central learning knowledge base that can be shared by all the users at the different factories, it is possible to collect as much knowledge as possible through the networks, acquire knowledge once and use it for all the factories. The repetition of the knowledge acquisition process will be reduced. With the use of this system, the knowledge base will grow and mature.

Based on this consideration, a web-based learning knowledge-based system is developed in this project. Multi-agent technology is used in the system. Learning and diagnostic agents (LDAs) are created to support multi-user remote diagnosis (working machines located in different places), and

learn new knowledge through the faults that occurred at all the sites. The central management agent (CMA) is in charge of updating the knowledge in the central knowledge base (CKB). The machine agent (MA) keeps records of the machine working status. These agents cooperate to realize remote monitoring and fault diagnosis, and on-line knowledge acquisition.

2. ARCHITECTURE OF THE MULTI-AGENT REMOTE DIAGNOSTIC SYSTEM

The framework of a remote fault diagnostic system with learning capability is shown in Fig. 1. According to the differences in location and functionality, this system is divided into three modules: the central management system (CMS), learning and diagnostic agents (LDAs) and the remote machine site (RMS). Each module is dedicated to some specific functions. Using agent technology in this system, these three functional modules are wrapped into three corresponding agents – the CMA, the LDA and the MA. CMS, central management system; CMA, central management agent; LDS, learning and diagnostic system; LDA, learning and diagnostic; RMS, remote machine site; MA, machine agent.

2.1 Machine Site

In this system, the machines to be diagnosed are located at different locations from the CMA. The machines are wrapped with the agent software into MAs. These MAs are in charge of sending requests to the CMA, sending real-time status signals to the DLA, and receiving messages from the other agents.

Theoretically, supposing the network transmission speed is fast enough, and the hardware and software can satisfy the response time requirements of the diagnostic system, the machines to be diagnosed on-line or off-line can be located around the world. For on-line fault diagnosis, it is essential to locate the machines inside an Intranet to ensure a quicker response. In this project, it is assumed that the network transmission speed is fast enough to allow globally distributed machines to be diagnosed remotely in real-time.

In this system, it is assumed that the overall number of machines at the remote sites is finite, and the types of machine are known. However, at any specific time, the number of working machines is not definite, and the corresponding LDAs will be initiated to implement the fault diagnosis and learning for these machines.

2.2 Diagnostic and Learning Agent

The module with the functions of fault diagnosis and learning is encapsulated into a LDA. In the prototype, one LDA is created to serve one machine at a specific machine site. Through communication between the remote machine and the DLA, real-time signals are transmitted to the LDA. The LDA

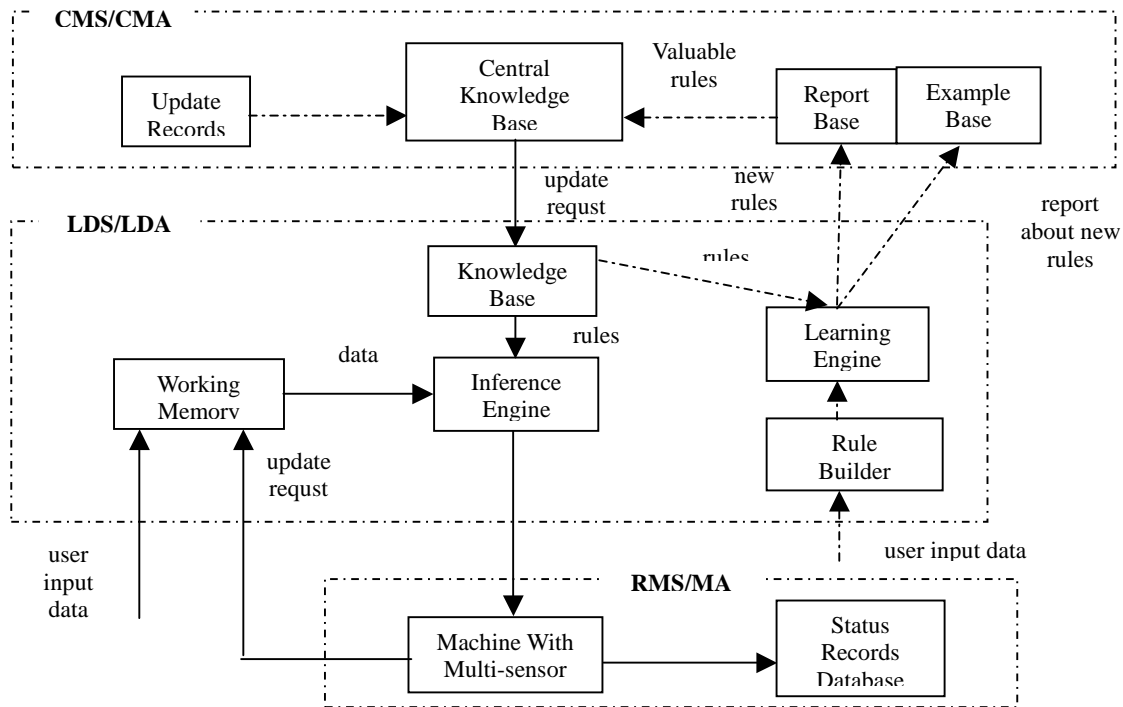


Fig. 1. System architecture.

CMS, central management system; CMA, central management agent; LDS, learning and diagnostic system; LDA, learning and diagnostic; RMS, remote machine site; MA, machine agent.

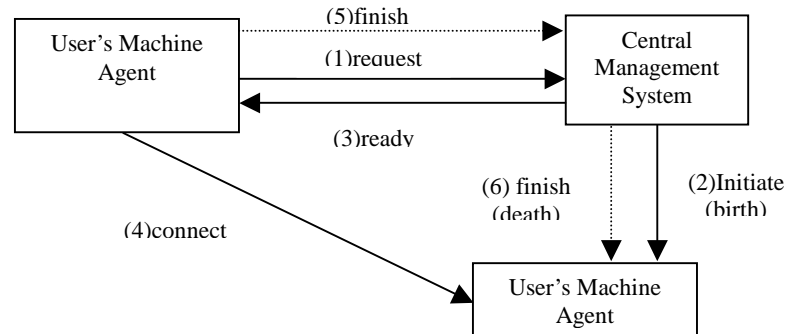


Fig. 2. Working process of this system.

monitors these machines and diagnoses the faults in real-time. At the same time, new knowledge is acquired using a learning algorithm through the fault diagnosis process.

2.3 Central Management Agent

The CMA consists of a central knowledge base (CKB), an example base (EB) and a report base (RB). The CMA has the responsibility for updating new rules into the CKB, and its related maintenance. After the learning processes have been implemented by the LDA, the newly created knowledge and the corresponding reports (explanations), from the learning engine in the LDA, are sent to the CMA, which stores them into its EB and RB. When the CMA receives a message from the LDA that there is no solution in the current LDA's knowledge base for a fault in a machine, the CMA will load new knowledge from the EB and send it to the LDA. Each time new knowledge is used in diagnosing a fault on the machine, the confidence factor for this new knowledge, which is a factor

to test the typicality of that knowledge, will be incremented by one count. This knowledge is added to the CKB in the CMA when its confidence factor is high enough. The knowledge base of the LDA is updated periodically with the new knowledge in the CKB. The system engineer sets an upper limit for the confidence factor. With the operation of this system, the CKB will grow and become very large. Hence, the long-term maintenance becomes an important issue for this system.

3. SYSTEM OPERATION

3.1 Working Process

Stationary agents are used in the prototype. The LDAs are located at different sites from the machines. The entire working process is shown in Fig. 2.

When any one of the remote machines in this network begins to work, the MA sends out a "request" message to the CMA.

After receiving a request from the MA, the CMA initiates a LDA for that machine, and sends the specific name of the LDA and a "ready" message to that MA. By calling the name of the LDA, the MA at the remote place establishes a connection with the corresponding LDA. The signals for the working status from the remote machine can be transmitted to the LDA in real time. The LDA monitors and tests the working condition of the machine on-line, to make sure that no abnormal condition appears, and the knowledge learned during the learning process is also transferred through the communication channels. When the machine finishes the jobs and is to be shut down, the MA sends out a "finish" message to the CMA, and the LDA changes into a "waiting" mode and waits for a new arrangement (new connection to some machines) from the CMA. Although the number of working machines at a specific time is not finite, the number of average working machines at a certain time can be estimated. Thus, the number of LDAs in the system can be set equal to the average number of working machines. If there are not enough LDAs, the CMA can initiate a new LDA easily (an inference engine, a learning engine and a rule builder are created).

3.2 Diagnostic Process

In this stationary multi-agent system, when a LDA is initiated, its knowledge base loads rules from the CKB in the CMA. Only the rules that are correlated to that machine are loaded. The diagnostic process in a LDA is shown in Fig. 3.

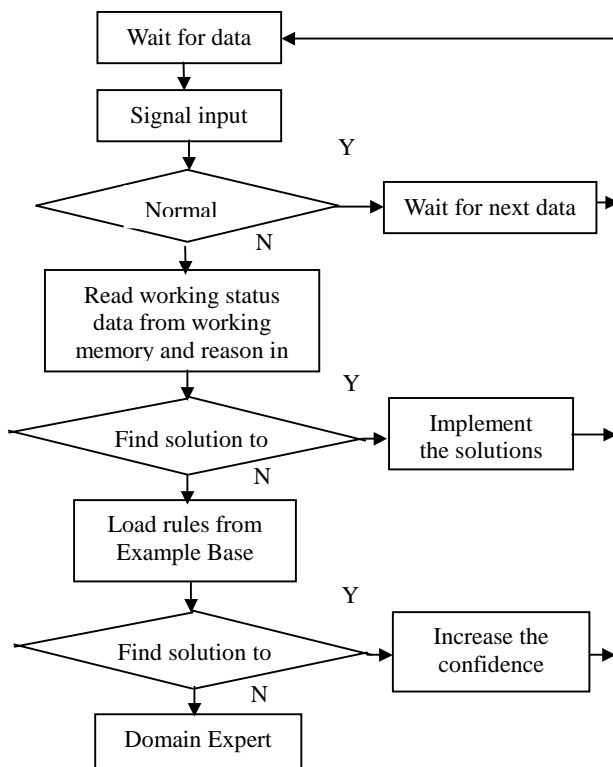


Fig. 3. The diagnostic process.

The LDA receives the status data from the machines in real time. These data are sent to its working memory. The inference engine loads the data from the working memory, and makes use of the rules from the knowledge base for reasoning. When the input signals are abnormal, the inference engine sends out an alarm message or some control signals to control the machine remotely and avoid the danger. If the fault is not new to the

knowledge base, this system will take action immediately to remove the faults.

However, it is possible that the faults are unknown to the diagnostic system, i.e. the rule base does not have sufficient knowledge to correct the faults. In such a condition, the LDA will request help from the CMA. The CMA will connect the LDA's knowledge base with the corresponding parts of the EB in the CMA, in order to load the related rules into the rule base in the LDA. The inference engine will reason again with the expanded knowledge base. An answer may be found using the added rules in the LDA. The corresponding confidence factor of the fired rule is increased by one count at the same time. After finishing these processes, the added rules will be removed from the knowledge base in the LDA and sent back to the EB.

3.3 Learning Process

The learning process of the LDA begins when a solution cannot be found using the rules in the EB. The fault is new to the diagnostic system, and the knowledge acquisition process must begin, which needs some help from the domain expert, in this prototype system. The learning process is shown in Fig. 4.

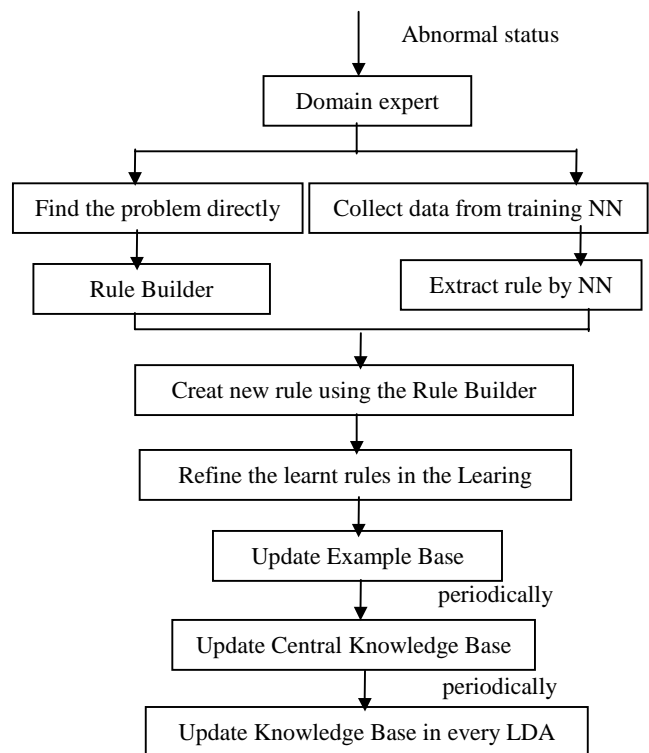


Fig. 4. The learning procedure.

The domain expert has two responsibilities: first, when the faults cannot be removed automatically by the system, the domain experts must go to the shop floor, check the records of the working status in the status records database, find the faults and propose reasonable solutions. After that, the domain experts assume the second responsibility: learning. The domain experts input the new knowledge in the form of a "rule" into the rule builder. The rule builder translates this original rule into the expert system language. At the same time, the domain experts should submit a short report to explain the new rule and give a detailed introduction to the fault and the solution. These documents are very useful for the maintenance of the CKB.

The newly created rules from the rule builder are of the same form as the rules in the CKB. However, these new rules may be partly repeated, or conflict with the rules in the CKB. With the rules provided from the rule builder, newly refined rules are created in the learning engine based on the rules in the knowledge base of the DLA, and thus the learning process in the DLA is completed.

4. SYSTEM IMPLEMENTATION

Java is used to develop the system, as it is a robust and reliable programming language that is platform-independent and Internet-supported. It is the best choice for developing a multi-agent system running on the Internet.

Sensors are necessary to monitor the working status of the machines, such as force, acoustic emission, power consumption, etc. The data to be transmitted for diagnosis is compressed before being sent out, in order to decrease the transmission loads. In practice, the diagnostic process may need more information about the machine status besides the signal inputs from the sensors. Multimedia technology may be used to acquire more information, for example, video-conferencing packages [1] can provide dynamic image transfer from the remote machines. In addition, in certain conditions, the operators at the machine sites can be asked to help provide the necessary information to the LDA directly through the agent telecommunication channel.

The transmission of the machine working status data is implemented in a TCP/IP client/server structure using Java. The machine works as a signal server, and a server socket is created using Java to send out signals from the sensors in real-time. On the DLA side, a socket is coded into the agent to receive the signals from the machine.

The learning function is an important feature of this system. As discussed before, there are many different algorithms to realize the learning of new rules, such as extracting rules from trained artificial neural networks [2], learning rules from experience using genetic algorithms[3], and extracting rules from databases using rough set theory[4]. In this project, a learning method based on the theory reported by Guan and Braham[5] is used. This method identifies all the possible paths of fault propagation for any possible failure sources through the representation of the physical connectivity of the devices under diagnosis. The structure is classified into many sub-devices that are interconnected. The interconnections between the sub-devices provide information on the possible paths of fault propagation, so that it is possible to trace the faults back to the responsible root causes of the faults.

The CKB is classified into three parts to include some of the common machining operations in this system. It includes the turning rule base, milling rule base, and the drilling rule base. Each rule base has its own rules for the corresponding machining process. When a machining process, such as the turning process, is initiated on a remote site, the CMA initiates a LDA (an inference engine, a learning engine and a rule builder are created). The knowledge base in the LDA loads all the rules from the turning rule base.

Each rule has a rule name, which is composed of a class name and an identification name. The class name is the rule base name, for example, rules in the turning rule base have the class

name TURNING. The identification name is the name that stands for the rule content. When a new rule is extracted through the learning process, its rule name is extracted from the report files. Every report file generated from the learning process, has the same class name and the same identification name as the name of the corresponding rule. The EB and RB are also classified into three parts: turning, milling, and drilling.

5. CASE STUDIES

5.1 Monitoring and Diagnosis of Tool Wear

In machining operations, the cutting tool usually performs under severe conditions of high temperatures (800–1000 °C) and high forces (2500–3000N). Gradually, the tool will lose its capability to produce the intended objective of cutting. Failures to detect tool breakdown may result in damage to the work piece and/or the machine and poor work piece quality [6–8]. Li et al. [9] recently reported a neural network model with fuzzy logic as a hybrid learning model for tool wear monitoring in drilling operations. The tool life can be estimated in different ways. Recent studies on the dynamic cutting force have shown that on-line tool wear monitoring may be feasible.

The onset of tool failure could be predicted by determining the threshold value of the percentage drop in the dynamic force, from its maximum amplitude, before the tool fails. This is a good indicator for predicting tool failure and it can be incorporated into the software program for on-line tool wear monitoring. In addition, the value of the dynamic force does not always increase monotonically with time or tool wear. These fluctuations of the dynamic force signals can be misinterpreted as a change in the trend when the effect may only be transient. The change in the trend of the dynamic force can be better assessed by setting a second condition that checks the gradient of the dynamic force curve. These two suggested conditions must be met to indicate the onset of tool failure.

5.2 Case Study

Currently, a simple prototype of the web-based knowledge-based system for tool wear fault diagnosis has been constructed. Two PCs were used to implement the system. The CMA is located on PC 1, on which the router is running. Here, the DLA was installed on PC 1. It can also be located on a different PC from the CMA. PC2 was used as the MA. These two PCs were connected through the Internet. The connection between these three parts is shown in Fig. 5.

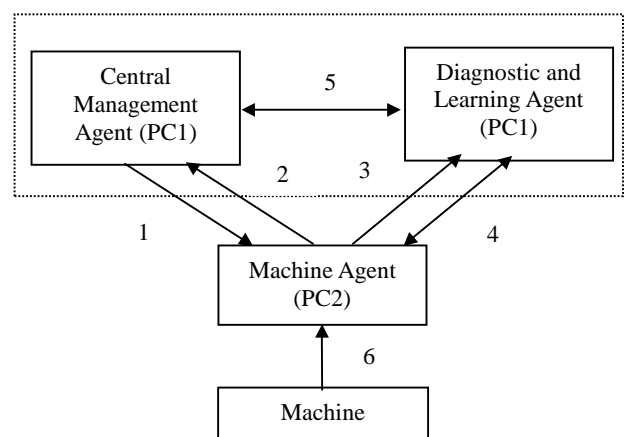


Fig. 5. Structure of the prototype system.

Sensors could also be used to acquire other information, such as temperature, and acoustic emission. The signals from the lathe were processed on the MA, and sent out through a server socket coded in the MA. The whole system was initiated by the following steps:

- Step 1. Run the Agent Message Router.
- Step 2. Initiate Central Management Agent.
- Step 3. Run the Machine Agent.
- Step 4. Ready for the Preparation.
- Step 5. Create Connection between MA and DLA.
- Step 6. Remote Diagnosing the Machine.

6. CONCLUSIONS

Case study demonstrates the capability of real-time on-line tool wear monitoring in the formulated web-based and agent-based architecture. Signals from machines located remote from the server are transmitted via the Internet to the server where the CKB is located for diagnosis, and solutions to these faults are transmitted in real-time to the remote machines to correct the faults. In the prototype developed in this work, the stationary agent technology is used. It is possible to apply the mobile agent technology in this structure. However, a compromise has to be made between the network safety and the network traffic. In addition, special requirements such as the central control of the subsystems should be taken into consideration in the design.

This system has several advantages. Through the central management system, the working conditions of all the distributed shop floors can be monitored and diagnosed in real-time. Located at the site of the central management system, the system manager can acquire the information of all the remote machines. Secondly, in this architecture, the shop floors share the same CKB. This saves the cost of maintaining the knowledge base at each site. The updating of the CKB is easier and less costly. All the machines are connected through the Internet, which facilitates the knowledge acquisition process. Each useful rule acquired from one site can be used by the other sites, which have similar machines. This takes advantage of the web-based technology in knowledge acquisition. Lastly, in this system, some methods are introduced to improve the maintainability of the CKB, such as modularity, graphical viewer, etc.

7. REFERENCES

- [1] P. Wayner, "New videophones starved for bandwidth", *Byte*, 21(5), pp. 125–128, 1996.
- [2] R. Andrews, J. Diederich and A. B. Tickle, "Survey and critique of techniques for extracting rules from trained artificial neural networks", *Knowledge-Based Systems*, 8(6), pp. 373–389, 1995.
- [3] F. J. Garrido and M. A. Sanz-Bobi, "Learning rules from the experience of an expert system using genetic algorithms", *Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, conference publication no. 446, pp. 226–231, 1997.
- [4] S. Tsumoto, "Discovery of rules for medical expert system-rough set approach", *Proceedings of Third International Conference on Computational Intelligence and Multimedia Applications*, pp. 212–216, 1999.
- [5] J. Guan and J. H. Braham, "An integrated approach for fault diagnosis with learning", *Computers in Industry*, 32, pp. 33–51, 1996.
- [6] T. Nakajima, J. Ahn and T. Sata, "Tool breakage monitoring by means of fluctuations in spindle rotational speed", *Annals CIRP*, 36(1), pp. 49–52, 1987.
- [7] S. B. Rao, "Metal cutting machine tool design – a review", *Transactions ASME, Journal of Manufacturing Science and Engineering*, 119(4), pp. 713–716, 1997.
- [8] K. Danai and A. G. Ulsoy, "Dynamic state model for on-line tool wear estimation in turning", *Transactions ASME, Journal of Engineering for Industry*, 109(4), pp. 396–399, 1987.
- [9] X. Li, S. Dong and P. K. Venuninod, "Hybrid learning for tool wear monitoring", *International Journal of Advanced Manufacturing Technology*, 16, pp. 303–307, 2000.



e-commence.

Pan Feng is an Associate Professor in School of Information & Control Engineering, Southern Yangtze University. He graduated from Southern Yangtze University in 1984. His research interests are in distributed processing, grid computing, network security and

A Design and Implementation of Dynamic E-business System Based on WEB Services

Shao-zhen Ye, Hua-Jun Han

College of Mathematics and Computer Science, Fuzhou University

Fuzhou, Fujian 350002, China

Email: yeshzh@vip.sina.com Tel.: +86-591-7803092

+86-13805054242(GMS Mobile)

ABSTRACT

Web Services is the focus of IT fields at present. The purpose of Web services is to provide a common technology layer that has no relation with language and platform. The application of different platforms, special in E-business system, needs this technology layer to connect and integrate dynamically with each other. Firstly, this paper introduces simple the concept and architecture of Web Services and analyzes the core technologies to construct Web services, XML, SOAP, WSDL and UDDI in detail. Secondly based on Web services and BPML4WS technology, an open and distributed dynamic E-business system structure is brought out and realized. Finally, the front design done on dynamic E-business system is summarized and further work is prospected.

Keywords: Web services; Dynamic E-business, BPML4WS, Intelligent Agency

1. INTRODUCTION

E-business is a kind of economic activity that makes use of advanced information technology and communication measure based on the network platform, and E-business also is the kind of new business mode that is different from conditional transaction one^[1]. In general, E-business includes all business activities through electronic transaction and computer network technology, for example market analysis, customer relation management, supplies dispenses, inner management and cooperation in the enterprise, etc. Along with business environment in the enterprise transformation continuously and market competition in the world, the business mode of enterprise has also take place greatly. In order to meet these requirements, there are many new organization forms between enterprises at present. Main forms have virtual enterprise by integrating many enterprises in different places and group enterprise by quick reorganization and dynamic alliance. Traditionally, the integrating application on enterprise to enterprise is mainly assumed as the needed solving blue print by reengineering single enterprise software, but this type of tight associative solving blue print is poor in agility. When business requirement needs altering, it's very difficult to resolve these new problems only by refitting this kind of integrated configuration. It is very clear that the E-business system existed could not adapt for the new business pattern today.

Dynamic E-business accelerates E-business's technology new development. Dynamic E-business provides such functions as business flow, dynamic extension and connection between clients and manufacturer systems for enterprises. By applying dynamic E-business, enterprises can search for business cooperation associates as soon as possible and establish

corresponding business affairs relations immediately. So it can accelerate the integration of enterprises and create more profits for enterprises. WEB services and BPML4WS provide effective support of technology for the implement of dynamic E-business.

Firstly, this paper introduces simple the concept and architecture of Web Services and analyzes the core technologies to construct Web services, XML, SOAP, WSDL and UDDI in detail. Secondly based on Web services and BPML4WS technology, an open and distributed dynamic E-business system structure is brought out and realized. Finally, the front design done on dynamic E-business system is summarized and further work is prospected. From work done, the distributed, open and dynamic E-business integrated system can efficiently resolve the problem of dynamic integration together with mutual operation existed in E-business.

2. SIMPLE INTRODUCTION OF WEB SERVICES

Web service technology occurs in accompany with the development of Internet technology, and belongs to distributed computing system. Web services are the on-line application service published by the enterprise for special business requirement, and other company or application software can visit and use these on-line services by Internet^[2]. In theory, any equipment such as PC, mobile phone or PDA can visit a variety of applications and services, and realizes automatically renew and update these applications and software at real time. In the implementation, Web services is packaged some operation interfaces such as service, process or method. Through these standard protocols, dynamic connection between programs can be realized easily. The most basic protocols include SOAP, WSDL, UDDI, and their basis is XML. From the point of users, Web services are some objects or components fixed at Web. They have many advantages such as good opening, high integrating, suitable standard, etc.

2.1 System Architecture of Web Services

The system architecture of Web services describes a framework. Through the framework E-business services can be described, published, discovered and called in the distributed computing environment. Three entities that realize services have real functions as follows:

- Services providers can publish and achieve services. In generally, provider shows some business function into a Web service that is used by other organization.
- Services requesters can discover and call services.
- Services registers can preserve the information related to services.

Fig.1 expresses three components how to communicate each other. Pay attention to that from different points the functions between these roles are not same^[3].

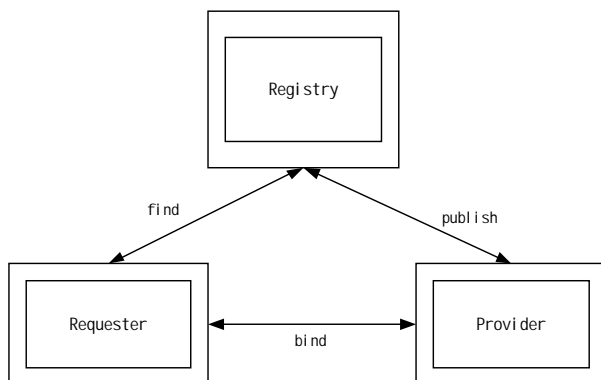


Fig.1 Web Services Model

2.2 Technologies Frame Of Web Services

In order to realize Web services, there is self-definite protocol inter OP stack in Web services system. Protocol inter OP stack is new protocol inter OP technologies. They mainly have XML, SOAP, WSDL and UDDI. That is as follows in Fig.2^[4]:

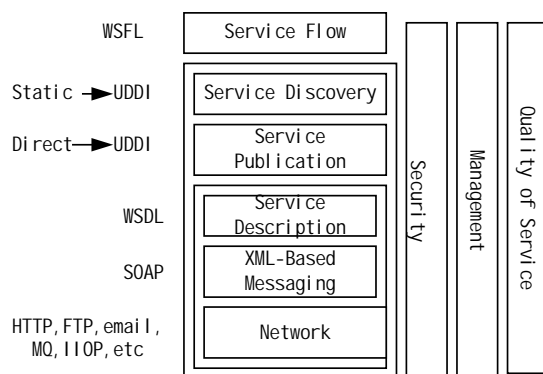


Fig.2 Web Services Protocol Stack

XML(Extensible Markup Language) is a kind of markup language that may be created into self-definition markup. So the markups in paper files can be given some meaning by XML. XML files are made of markup, element and attribution. XML exchanges data simply and support intelligent search.

SOAP (Simple Object Access Protocol) is simple protocol that is used for communicating information in the distributed environment. It is based on XML protocol, and contains four parts: SOAP envelop defines that what is its content in message depiction; who sends the message; who should accepts and deals with it; how to deal with their frame. Though the four parts are defined into the whole group^[5]. But their functions are interactive and unaided. Its two main design goals are simple and expanding.

WSDL (Web Services Description Language) provides a kind of grammar that can be described service into a group of information exchange port^[6]. WSDL files are a kind of depiction that is not relate to language and platform. WSDL describes services, visiting method, expecting response pattern. WSDL files can be exchanged through private or UDDI

register center. And WSDL is also file format that is based on XML. It is used for pattern describe, message, operation, interface, position and protocol retained.

UDDI(Universal Description, Discovery, and Integration) provides a kind of middle system that is used for published and located service describe. UDDI supports different service definition patterns such as WSDL files, standard JAVA interface and XML files. UDDI describe all API of register center. The API finishes two basic tasks: register enterprise and service, locate and bind a service registered. Register and location is finished by means of UDDI command being put in body of SOAP message and sent into register center. The basic system architecture of UDDI register center node is showed in Fig.3 In fact, UDDI center may publish Web service, which can be browsed by clients through register port in the UDDI center. UDDI also support three kinds of data in register center, white pages, yellow pages and green pages. Fig.4 describes a variety of UDDI data and their relation.

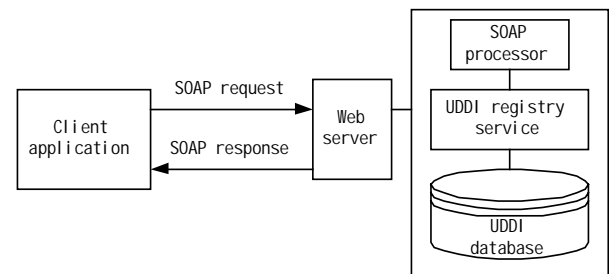


Fig.3 the Basic System Architecture of UDDI Register Node

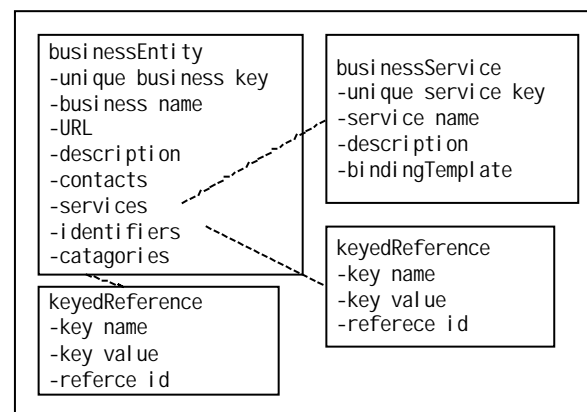


Fig.4 UDDI Data and their Relation

3. THE DESIGN ON E-BUSINESS SYSTEM ARCHITECTURE

Because WDSL only supports interactive model that is at no state, it is not suitable for describing equal messages in state existed. In order to enlarge Web services interactive model and support business flows, BPEL4WS (Business Process Execution Language for Web Services) provides a formal language that is used for expressing business and business inter-protocol^[7]. BPEL4WS has two basic concept methods: one is abstract flow concept that may definite business protocol role; the other one decides inter-protocol by means of definition of business flow doing. With the help of BPEL4WS and Web services frame, an open and distributed dynamic E-business system structure is brought out. The structure can

build up easily business flow application program that is not related to platform, and is distributed and agility.

a) Frame Model

Dynamic E-business model architecture is as follows Fig.5. The system specially emphasizes two important aspects: role and work fashion.

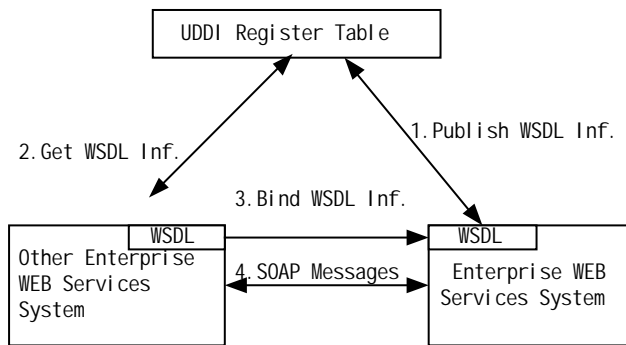


Fig.5 Dynamic E-business Architecture Model

Main functions in the system are:

1. Enterprise Web service system takes business flow to express into a kind of Web service, which is called by other enterprise. Then publishes enterprise information, which includes business and Web service technology standard into UDDI register tables in WSDL file format.
2. As the client, Web services system of other enterprise may discover Web service that is useful for itself. That is that search UDDI register tables, find business and its service, download technology standard in WSDL files.
3. Client decides communicating and binding information with WEB service communication of services system.
4. Client port communicates with Web service system of enterprise through SOAP protocol.

b) Implementation

According to the architecture frame in the Fig.5, for inter information functions in the enterprise should be accomplished by the enterprise WEB Services System. Which contains message processor, receiver and sender based on role, intelligent media based on client role, intelligent media based on role called, role joined and real program. The system frame in detail is showed in Fig.6. These special functions in the system are introduced as follows.

--Messages Processor, after being unzipped and decrypted, SOAP message is sent into the transmitter based on role by HTTP protocol.

--Transmitter Based on Role, After SOAP message is analyzed and sent into related role process module according to the role of visitor.

--Intelligent Agent based on client role, Its basic function contains unzipping processing; deciding caller identifier; deciding caller security book; identifying caller authorizing information; deciding caller environment information; identifying caller data complete; deciding caller service request; dialogue policy management; central controlling kero; adding environment information; etc.

--Business Flows, Its function contains the business trade

between enterprise and enterprise and the business process within enterprise.

--Intelligent agent based on role called, in fact it is the middle exchange post between role called and E-business system. Its function includes adding environment information; adding security book; adding identifier's information; zipping process; dialogue policy management; etc.

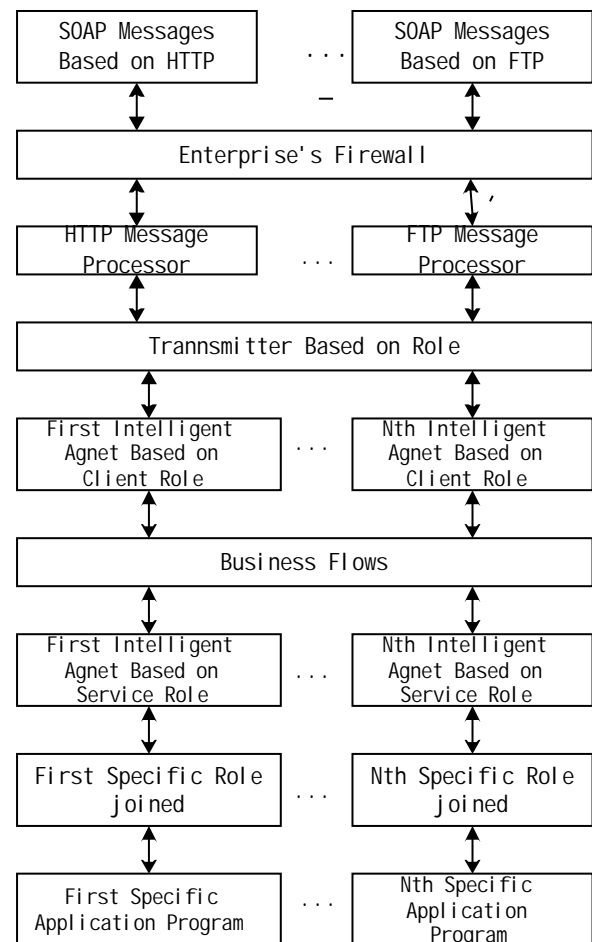


Fig.6 The function module frame

c) Application Example

With the help of Web services for J2EE, Apache AXIS frame that is the newest SOAP standard from Apache Group and based on Java and provides with opening original code according to SOAP with Attachments standard. For the execution BPEL4WS file, BPWS4J(IBM Business Process Execution Language for Web Services Java Run Time) is used for the work flow agent and provided with software service environment for the course example. In fact, BPWS4J concludes two basic components, which are the platforms executed and Eclipse plug-unit. And BPWS4J is the implementation about BPEL4WS standard. Through the tool, the editor that is used for building up the flow of BPEL4WS will be integrated into the development environment. Now with means of a variety of WEB services and the development technologies mentioned before, the application implementation about some real example of E-business is introduced. The main object of this example is as follows and showed in Fig.7.

--The client on Internet may be permitted to hand in some order into the E-business system.

--The detector will be chosen automatically according to the order of client and validated a variety of information of the order.

--According to the validating reason, the calculator is selected out and calculates the price of order.

--The evaluator is chosen to judge and the reason is sent into the client.

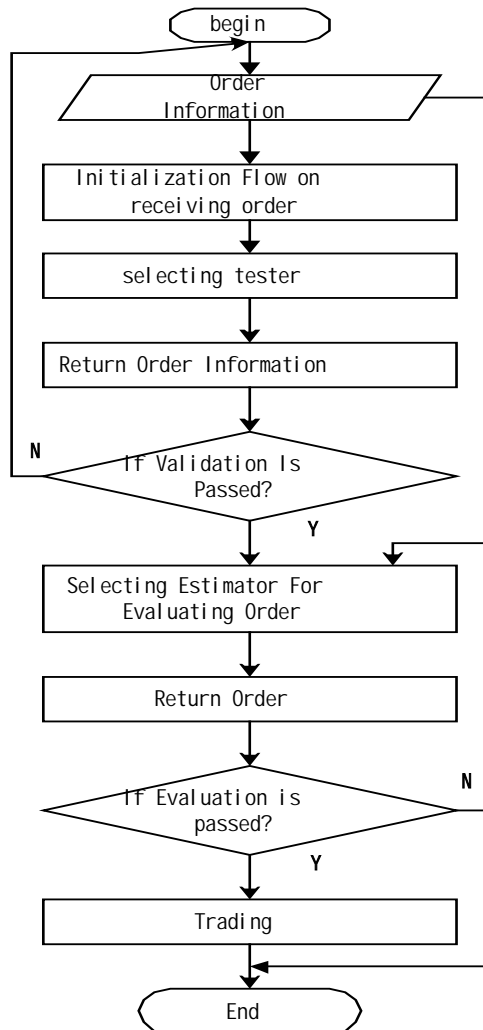


Fig. 7 The Business Flow

Though the E-business flow chart is not complicated, some problems on the specific implementation under the development of BPEL4WS mounted need to be solved. Firstly the business flow is related to five business roles such as buyer, seller, validator, calculator, and evaluator. These service roles are described as Table1. The description of serviceLinkType is used for expressing inter actions between roles. The flow chart may be showed as Tab.2.

Tab.1 Service Roles Description

```

<partners>
  <partner name="Seller"
    xmlns:ns1="urn:EbusinessService"
    serviceLinkType="ns1:SellerSLT"
    myRole="salesSystem"/>
    <partner name="Buyer" xmlns:ns2="urn:
    EbusinessService" serviceLinkType="ns2:BuyerSLT"
    myRole="buyerSystem"/>
    <partner name="ValidationService"
    xmlns:ns3="urn: EbusinessService"
    serviceLinkType="ns3:ValidatorSLT"
    partnerRole="validator"/>
    <partner name="CalculationService"
    xmlns:ns4="urn: EbusinessService "
    serviceLinkType="ns4:CalculatorSLT"
    partnerRole="calculator"/>
    <partner name="AssessmentService"
    xmlns:ns5="urn: EbusinessService"
    serviceLinkType="ns5:AssessorSLT"
    partnerRole="assessor"/>
</partners>
  
```

Tab.2 The business Flowchart

```

status=(time:0 type:-2 role: Buyer)
while (status:type=-1 or status:type=-2)
  receive Buyer:create Order //等待创建一个新
  订单的请求
  switch
    case status:type!=-3
      invoke ValidationService:
      validate //switch 结束
  switch
    case status:type=1
      invoke
      CalculationService:
      calculate//switch 结束
  reply Buyer create //向客户返回订单信息
//循环结束
switch
  case status : type=2
    receive Buyer : agree//等待同意订单的请求
    switch
      case status : type=3
        invoke AssessmentService :
        isCreditable// switch 结束
        reply Buyer : agree// switch 结束
    switch
      case status : type=4
        receive Seller : approve
  
```

The function of intelligent agent may be realized into WEB service by AXIS frame. The real implement is that the core function is by means of WEB services and the other functions are by means of Handler function of AXIS. On the environment of AXIS, SOAP message must realize the interface of org.apache.axis.Handler. AXIS may define the service and processing programs on server by

server-config.wsdd file. For the intelligent agent based on role called, E-business system must firstly visit the intelligent agency, secondly according to the message processed by the intelligent agent based on client role and the client need, the business partner may be dynamic chosen. The information client needs may put in the user setting information. The setting information exists in XML file format. The class of service provider of the business partner may refer to the provider given by <serviceprovide> class.

The locators are classified as three classes, static; UDDI; mobility

--static, the special service implement is located by service attribute that refers to the WSDL definition of service;

--UDDI, The service implement is discovered by requiring about API in UDDI SOAP according to the register of tModelKey researching. Of course, there are many kinds of methods on location.

--mobility, the service provider is used in the message produced by action lying in flowchart.

The publish of intelligent agent based on role is realized by WSDL file. The WSDL file is showed in Tab.3.

Tab.3 the WSDL File of Buyer Role

```
<definitions name="Buyer" ..... >
.....
  <portType name="BuyerPT">
    <operation name="create">
      <input message="def:myOrder"/>
      <output message="def:myStatus"/>
    </operation>
  </portType>
  <binding name="SoapBinding" type="tns:BuyerPT">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="create">
      <soap:operation soapAction=""
style="rpc"/>
      <input>
        <soap:body use="encoded"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
namespace="Buyer"/>
      </input>
    </operation>
  </binding>
</definitions>
```

4. CONCLUSIONS

Dynamic E-business is the next new one that focuses on inter-action between service objects. Whose technologies and implements can bring a great business opportunity to enterprises. Dynamic E-business theory related to technologies on WEB services and BPEL4WS have been researched and discussed in this paper. And based on the work done before, A system architecture of dynamic E-business is brought out and main implements on Web services is also given simply. All the front work will be in favor to research and develop more complex E-business application system in the enterprise of China.

The future work is that we focus on the foreign trade flowchart in the enterprise of China. With the support of 863 project in China, special in foreign business department of Fujian Furi Electronics Company Ltd, the E-business application system based on foreign trade business in enterprise by means of J2EE factual standard and Web services.

5. REFERENCES

- [1] Ming Qi, Wei-long Yan, Yu-zhong Zhai. The E-business practical course. Beijing: high education publish house, in 2000, pp.4~8
- [2] Holt Adams , Dan Gisolfi , James Snell , Raghu Varadan.the best practices on Web service: back to basic part. In 2000. <http://www-900.ibm.com/developerWorks>
- [3] Heather Kreger , the conceptual system architecture on Web service, in 2001. <http://www-900.ibm.com/developerWorks>
- [4] Doug Tidwell , Web services—the next revolution on Web, in 2001. <http://www-900.ibm.com/developerWorks/>
- [5] Zhi-hua Duan, the basic talk on SOAP, in 2001.<http://www-900.ibm.com/developerWorks/>
- [6] Xiao-lu Cai, WSDL: description on your Web services,in 2001. <http://www-900.ibm.com/developerWorks>
- [7] Francisco Curbera , Yaron Goland , Johannes Klein , Frank Leymann , Dieter Roller , Satish Thatte , Sanjiva Weerawarana, the business chart languages of Web services 1.0 , in 2002. <http://www-900.ibm.com/developerWorks>

附中文参考文献：

- [1] 祁明,晏维龙,瞿裕忠等,电子商务实用教程,北京:高等教育出版社,2000.4~8



Ye Shaozhen is a Associate Professor and a deputy director of department of Computer Science and technology, the college of mathematics and computer science, Fuzhou University. She graduated from Fuzhou University in 1984 and got bachelor's degree. And got master's degree majored in Computer application from Fuzhou University in 1995. From 1998, she has been on the job and read for doctor's degree majored in computer application of Tsinghua University. In 2004 she got doctor's degree majored in Computer application from Tsinghua University.

Research and Design for the Wrapper of Web-Based Data Resource Assembling Based on Soap

ZhiHua Li Jun Sun

School of Informaiton Technology, Southern Yangtze University
Wuxi, Jiangsu 214036, China

Email: zhli@sytu.edu.cn Tel: 0510-5863643

ABSTRACT

In this paper, we introduce briefly the web database, web-based data resource and Soap protocol, and describe the accessing model of heterogeneous web data resource assembling. And therefore, after studying the inter-operation on heterogeneous components, we analyze the approach of distributed accessing to web-based data resource of Soap, and present the accessing model. In the last section of the paper, we propose a designation of a wrapper and its paradigm using Java.

Keywords: Soap, Data resource, Distributed accessing, Wrapper

BACKGROUND

Nowadays, web technology has become essential to software development, network distributed computing and so forth. Relative to the data of related database with strong structure, every network station on the web is a heterogeneous data resource, and therefore, all of the network stations on Internet comprise a vast heterogeneous database environment, the data of which is semi-structural. The database alike is called web database. In such a environment, the data has various forms, such as related data, structural file, multimedia data, data of object-oriented database etc. It is evident that data assembling, digging and eclectic exploiting based on web is far more complex than those data based on single related database. In this paper, we present our research on the wrapper, an application web database.

1. THE FEATURE AND SHORTCOMINGS OF CURENRT WEB DATA RESOURCE

a) Heterogeneous of data environment:

The heterogeneous of the database includes the heterogeneity of many aspects, such as the platform where the data is based on, the system environment and the internal data structure. To the users, the heterogeneity of information resource lies in the interface and means to access to different information resources. Thus, there are two technological problems due to the heterogeneity. One is how to assemble the heterogeneous data; the other is how to query the data on web.

b) The semi-structural data model

The traditional database has its own data model, according to which the data can be described concretely, while the data on web is not entirely structural because of its relative independence, self-depiction and dynamic changeability. So we call the data on web semi-structural.

c) The similarities and differences between assembling of web data resource and assembling of heterogeneous database.

- i. The formality of the two is similar.
- ii. Web data resource has vast increasing amount of data to dispose.
- iii. There are less metadata depicting the characters of the web dataresource, but it need much metadata to dig, query and exchange the data.
- iv. Each web data resource has strong autonomy.

2. THE MODEL OF ASSEMBLING SYSTEM OF WEB-BASED DATA RESOURCE

There are two approaches to assemble the web information—warehousing model and virtual model. This paper is only involved in virtual model, which is depicted in figure 1.

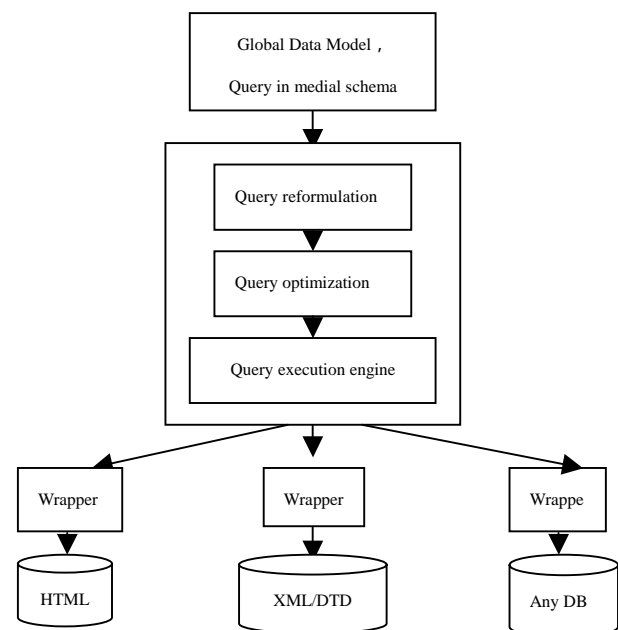


Fig.1 The integration model about data resource of heterogeneous Web

The virtual model is based on a media schema, in which the data is stored in a local data resource. The data is virtualized into the media schema by the data resource wrappers, and the users query the data through the wrappers without knowing the characteristic of a special data resource. And then, the media schema will be exchanged into the modal based on local data resource. In this model, one of the technological problem, which we will discuss in this paper, is how to design

the wrappers.

3. DESIGN OF THE WRAPPERS BASED ON SOAP

3.1 A Brief Introduction to SOAP

SOAP (Simple Object Access Protocol) is a data sharing protocol under distributed computing environment based on XML, and it is a protocol of Layer used display independent of transferring protocol, by which the data can be exchanged in form of object among the applications. SOAP itself does not define any semantics of application, and it provides a package model with standard component and the mechanics of data encoding in the modular. It also define a mechanics to display the semantics of applications. Due to its perspicuity and scalability, SOAP can be applied in the distributed system conveniently.

There are two communication method of SOAP, SOAP request and response. SOAP consists of a wrapping structure, a integral framework, a mechanics that define the content of a message and determine who can dispose the content and whether it is obligatory or not. Besides the SOAP wrapping, SOAP encoding rule and SOAP RPC protocol, SOAP must solve the problem how the message contained in the http message to send out.

3.2 Heterogeneous Web-based Data Resource Accessing Model Based on SOAP

As for the weak coupling relationship among webs, it is conformed to its charactersitics to adopt a simple emboking depiction approach. SOAP is entirely based on SML, and it

inhere the scalability and depictability. In a work, the encoding rule of SOAP define a series of mechanics of the paradigm employed to exchange the datatype defined by applications. Combined with distributed computing technology, such Windows DNA, accessing to distributed data resource can be easily implemented. The model is shown in figure 2.

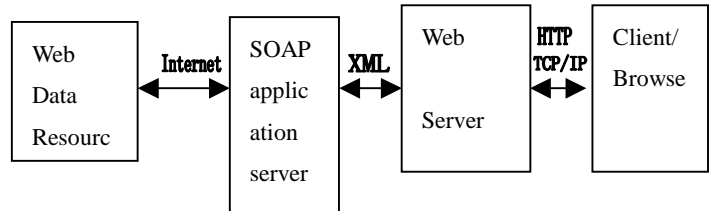


Fig 2 The integration model about data resource of heterogeneous Web

In this application, web server works as SOAP client end to response the request of the client end and change it into the SOAP request of SOAP application server. The designation of the SOAP server is very complex, and the wrapper is realized on the SOAP application server.

3.3 Wrapper DesignBased on SOAP

The main function of the wrapper is that it can exchange a interface into another interface that as users need, without alter the previous interface, in order to realized the inter-operation between the heterogeneous distributed component. The wrapper can make incompatible component work together to realize the accessing to web-based data resource under distributed environment. Figure 3 show the method of accessing heterogeneous component platform on client end.

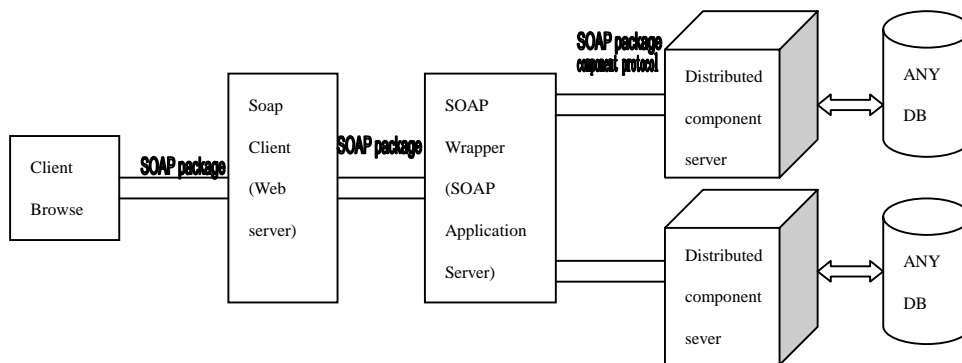


Figure 3 Accessing heterogeneous component platforms on client end

In this model, distributed component sever works as the client end of soap application server. The wrapper realized on distributed platform lies on the SOAP application server. Its main function is responsibility to emboke related distributed component by standard SOAP protocol. ANY DB can be data of any form theoratically.

3.4 Inter-operation among Heterogeneous Distributed Ccomponent

Based on the accessing model shown in figure 3, it is vital for inter-operation among the components. Inter-operation will execute the following task.

- Http is a protocol for transferring plain text and its dataflow can pass through 80 ports smoothly, whereas SOAP is a XML-depicted protocol and can combine SOAP with Http.

- Define an integrated DTD or Schema to analyse correctly the XML file embedded in SOAP package in order that the correct emboke to a particular component can realize.

- Develop Soap client end request application and SOAP sever end response application. Realize receive and dispose correctly SOAP package and the outcome of user's emboke to dipose distributed component.

The following gives two exampleof SOAP request and SOAP response:

- Emboke the SOAP client end request of distributed components

```

Post/StockQuote HTTP/1.1
Host: 202.195.144.148
Content-Type: text/xml;
  
```

```

charset="utf-8"
Content-Length:xxx
SOAPAction:"http://202.195.144.148/soap/mydbEJB.sql"
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop"

  SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/so
  ap/encoding">
<SOAP-ENV:Body>
  <n:ADBMethodInEJB
xmlns:n="java:comp/env/ejb/AccessDBEJB">
  <parameter>....</parameter>
  </n:ADBMethodInEJB>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

The response package of SOAP sever end to above request

```

HTTP/1.1 300 YES
Content-Type:text/xml;
charset="utf-8"
Content-Length:xxx
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelop/
"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/
encoding"/>
<SOAP-ENV:Body>
  <n:ADBMethodInEJB
xmlns:n="java:comp/env/ejb/AccessDBEJB">
  <result>....</result>
  </n:ADBMethodInEJB>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

4. CONCLUSION

Assembling and digging of web-based data resource is a frontier research field. It must provide users for transparent access. There many opened problem need further exploration, such as how to mask the conflict of data resources and semantics, discern different naming method and so forth. Our papers only involve the realizaiton of one aspect of these problems, and the rest is our research focus in future.

5. REFERENCES

- [1] DuCharme B: A Simple Soap Client <http://www-105.ibm.com/developworks/>
- [2] Fernsnde M,et al.XML Query Data Model.W3C Working Draft, Feb.2001
- [3] XML and Database A Database Techniques for the World-Wide Web:A Survey.SIGMOD Record,1998,27(3)

Li Zhihua is a Full Associate Professor, and engages in School of Information Technology, Southern Yangtze University. He graduated from Southern Yangtze University with specialty of computer's applications in 2000; He has published over 10 Journal papers. His research interests are in network security, artificial intelligence, network application, distributed and parallel processing.

An Approach towards Automated Web Services Composition*

Muhammad Adeel Talib & Yang Zongkai
Department of Electronics and Information Engineering,
Huazhong University of Science and Technology,
Wuhan - 430074, Hubei, Peoples Republic of China.
Email: adeeltalib@hotmail.com Tel: 086-027-62957910

ABSTRACT

With the growing number of Web services, importance of composing existing Web services into more complex services in order to achieve new and more useful solutions is increasing. A landscape of languages for Web services composition has emerged and is continuously being enriched with new proposals from different vendors and coalitions. However, current approaches based on these languages are rather restricted and inflexible as they lack proper support for generating dynamic compositions, which is a major challenge in this new paradigm. In this paper we present an approach to facilitate automated Web service composition in Service-Oriented Architectures using business rules. It is our belief that business rules can be used to determine how a composition should be structured and scheduled, how the Web services and their providers should be selected, and how service binding should be conducted. This paves the way towards developing automated Web services compositions.

Keywords: Web services, Composite Web services, Dynamic composition, Business rules, composition framework.

1. INTRODUCTION

Web service composition is about the creation and provision of complex value-added services from individual services*. Creation and provision of complex services comprise of composition/orchestration, aggregation, and coordination of partial services and providers. The main problem is to find a well defined environment that allows creating compositions efficiently. Generally the environment in which services are composed can be categorized into i) Static composition and ii) Dynamic composition.

In a static composition, the services to be composed are chosen at design/built time i.e. the control flow and data flow amongst the component Web services are given by the user, while in a dynamic composition, the services are chosen at run/execution time. The control flow and data flow amongst the component Web services are generated automatically. If the process to be composed is of a fixed nature wherein the business partners/alliances and their service components rarely change, static composition will satisfy the needs, but this usually is not the case. Following are the reasons:

Firstly, the global economy is volatile and dynamic. Organizations are changing constantly, entering into new markets, introducing new products and restructuring themselves through mergers, acquisitions, alliances and divestitures. New laws are levied. Businesses have to accommodate changes in applications, technology, and

organizational policies. A static process is not admissible. Secondly, increasing numbers of interesting services are moving online and the web is rapidly transforming from a collection of static pages to a provider of numerous useful services. Variety of services has to be compared and a vast space has to be searched. Services come and go, new services appear, existing services may be removed, servers are down, services may become temporarily unavailable, better services replace rotten services, functionalities and quality of services keep changing. This dynamic environment mandates automated service composition that can promptly adapt to the changing environment.

The challenge is therefore to provide a solution in which dynamic service composition development and management is facilitated in an automatic fashion. In this paper we attempt to propose a framework for automated Web service composition using simple business rules.

The rest of the paper is organized as follows. First we provide a common example that elaborates the motivation of Web services composition in businesses (section 2). Then we define and discuss the benefits of using business rules in order to orchestrate services into a composite function (section 3). Related research in the area of rules based Web service composition is described in section 4. Section 5 presents our proposed framework and then in section 6 we point out some requirements for composition and illustrate how our framework keep up with these requirements. Finally, we conclude by discussing future directions in section 7.

2. MOTIVATING SCENARIO

Consider a Manufacturer's supply-chain-management scenario as shown in the Figure 1 wherein the manufacturer exposes a process as a composite Web service that provides order price and delivery details to its customer (a distributor). When the distributor places an order, the manufacturer checks the inventory to verify if it has enough goods to satisfy the order. In case there is enough stock then the manufacturer contacts its delivery partner for a date of delivery and its accounts department to fix a price for the order. Based on the price returned by the delivery partner and the fix price methods, the price for the order is finalized. Then, the delivery date and the finalized price are returned to the distributor. In the other case, when there is not enough stock in the inventory, the manufacturer contacts its supplier partner for the required components. Then the manufacturer contacts its delivery partner to arrange for delivery of components to its manufacturing site and later to deliver the products to the distributor. The price and delivery date are returned to the distributor.

All the tasks involved in this process can be represented as Web services. The whole process is in fact a composition of

* In the remainder of this paper the word "service" and "Web service" will be used interchangeably.

individual Web services. This typical process depicts the vision of Web services composition – seamless integration of service request and response across organizations. However, the process is prone to changes that may frequently occur varying from domain to domain. For instance, a need might be developed in the future to add some more activities like checking the user's credit or invoking a function from backend legacy system. With time the manufacturer can switch supply partners due to better offers. It can merge with a delivery partner thereby eliminating other delivery partner services. There could be a change in the flow from sequential to parallel. We believe that all these changes can easily be incorporated into the system if the system is governed by business rules.

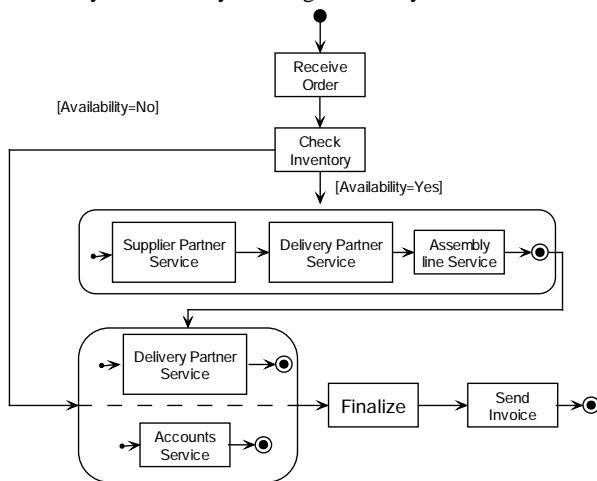


Figure 1 SCM Scenario

3. RULE BASED COMPOSITION

Business rules are precise statements that describe, constrain and control the structure, operations and strategies of a business [1]. They provide a way to specify business logic and have the advantage of combining automatic executability with a relatively high level of human understandability, i.e., a high conceptual level and a "declarative" (rather than only procedural) semantics like that of composition languages. The level of abstraction of rules facilitates their expression in user-friendly languages such as controlled natural language. This enables non-programmers, especially business-domain experts such as IT managers, to specify business rules, and to modify them relatively easily and often without knowing technical details.

The use of rules in service composition has been documented in a number of publications [1-4]. The authors claim that rules are more concise and easier to understand, share and maintain, especially in a global open environment such as the web, where self-documenting specifications are one of the current approaches to enable interoperability. The most important benefit of using this approach in Web services scenarios is that it allows automated composition without using complex formalisms. Fundamentally service composition is a business process that needs to be integrated with business events, business constraints, policies, strategies, and regulations. These parameters are the basic entities in a business application that are prone to frequent changes. Therefore these must be separated from the application code in order to be more easily managed, defined, verified and consistently executed. Business rules provide a means to do that.

4. RELATED WORK

Current composite Web service development and management solutions are very much a manual activity. Manual in the sense that user is expected to setup the complex workflow at XML level. This requires the user to have low level knowledge of composition languages which are quite non-trivial to grasp [5]. This applies even to applications that are being developed on the basis of available standards, such as BPEL4WS [6] or BPML [7]. With the growing popularity of BPEL (short for BPEL4WS) numerous vendors are offering their composition products based on the language. IBM's BPWS4J [8] provides a graphical interface to design composite service flows. Oracle's BPEL Process manager [9] provides a more intuitive graphical drag and drop interface, but still the user has to know the constructs of the composition language to design the process. To the best of our knowledge none of these solutions provide automated composition, and also, these GUIs are not feasible for large complex workflows.

The SWORD toolkit [2] automates service composition by using rule-based service descriptions. The user specifies the initial and final state facts. Based on this, a planner attempts to put together a chain of services that can satisfy these requirements. It mainly focused on the composition of information providing services (i.e., not world-altering services). The user is expected to be able to specify the state facts. The component services taking part in the composition must be known in advance. Main shortcoming is that it uses its own description language and does not support any existing standards like WSDL which makes the composition isolated and non-sharable.

DY_{flow} [3] is a dynamic Web service composition system developed at University of New South Wales, Australia. It is capable of dynamic composition and modification of running workflows by using a business rule inference engine. Two kinds of rules are defined viz. service composition rules and service selection rules that steer the composition engine in generating workflow schemas at runtime. The service composition rules are in the form of backward chain rules, forward chain rules, and data flow rules which indicate pre-conditions, post-conditions, and data flows respectively. Statecharts are used to model the process flows and an XML based language is developed to map the process specifications into the composition engine. The approach has more emphasis on dynamic process execution and management in production life-cycle where the rules have to be defined by domain experts.

Dynamic Workflow Model (DWM) [4] is a part of information infrastructure for supporting Internet-based Scalable E-business Enterprise (ISEE) developed at University of Florida. It extends the underlying model of the WPD (Workflow Process Description Language) of WfMC by adding event, trigger and rule to WPD's modeling constructs. Activities in the process can post three kinds of events i.e. Before-Activity-Event, After-Activity-Event, and External-Event. Business rules are attached to these events by using trigger specifications. These rules have the format Condition-Action-AlternativeAction. All the tasks in the business process are predefined and the rules cannot be modified dynamically. The work is focused on inter-organizational workflow management.

Oriens [1] proposed a model driven architecture for Web

service composition wherein a phased approach to service composition is used consisting of definition, scheduling, construction, and execution phase. The so called Information Model is based on constructs such as activities, messages, conditions, events, etc having special attributes. These constructs represent the required information for the composition. Business rules are then used to link, associate, and relate the constructs into a full fledged composition. The rules seem to be rigid and inflexible. The framework does not explain how the designed composition will be mapped into an executable form. Also other important details like co-relational dependencies among the services and service binding with concrete services are not defined clearly. We base our work on this approach and attempt to remove the deficiencies.

Much research on Web services composition is being going on in the semantic web community as well. They focus on reasoning about Web services by explicitly declaring their preconditions and effects using terms from pre-agreed ontologies. A semantic language is used to define the preconditions and effects like RDF or OWL. Artificial intelligence techniques are then used to synthesize individual services given an initial state, an explicit goal representation and a set of possible state transitions. Unfortunately semantic annotation of Web services does not exist so far in practice and neither there is motivation for Web services providers in doing that [10]. Also ontology design is a skill that is not widely found in the workforce [11]. Current tools, such as *Protégé* provide only limited help, and they have not been widely used outside of prototyping projects and research groups. Our work does not rely on any semantic markup of services.

5. PROPOSED FRAMEWORK

Overview

We believe that business processes can be built dynamically by composing Web services in a model driven fashion where the design process is controlled and governed by a series of business rules. In current web service technology solutions such rules are deeply embedded in the implementation of the processes, leaving the user with little empowerment to manage and control them and eventually the processes themselves. We propose a framework based on business rules to develop service composition that separates the abstract definition of process from the executable form. The framework architecture is shown in Figure 2.

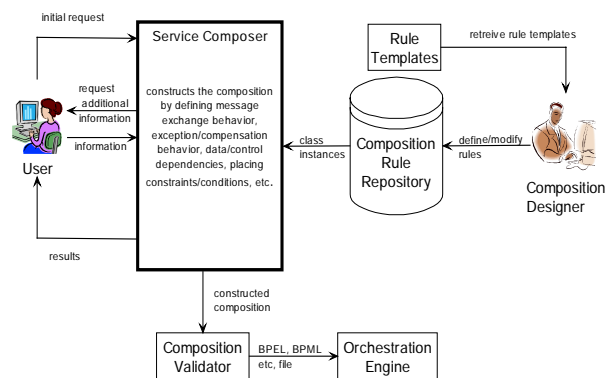


Figure 2 Proposed Framework

The information required for developing service compositions is identified as business rules that are represented in the form of classes having special purpose attributes that capture the information. These classes are modeled as *Rule Templates*. The information (business rules) gathered with the help of these templates is stored in a *Composition Rule Repository (CRR)* in the form of instances of classes. This composition rule repository is then used by the *Service Composer (SC)* to govern the development of service compositions based on a *Composition Algorithm*. The composer constructs the composition in a standard composition language (BPEL in our case) and sends it to the *Composition Validator (CV)* which verifies and validates the composition generated by the composition engine. Ultimately the validated composition is passed on to a commercial *Orchestration Engine (OE)* which executes it.

It is the duty of the composition designer to capture the domain specific details about the activities involved in the composition, the messages to be exchanged, possible exceptions, the role responsible to perform the activities, and the service providers. This can be done in a piecemeal fashion by progressing from abstract service description to more concrete implementation i.e. the developer can first concentrate on the activities involved in message exchange behavior and the possible exceptions without worrying about the service bindings which can be later incorporated into the system.

Information required for composition

Thorough study of composition languages reveals the fact that a composition is piled on some basic building blocks that capture each and every detail. "Message" is the fundamental building block of service composition. In other words, a whole business process can be expressed as an exchange of messages between two or more activities. The message may optionally have a payload made of a primary business document. Exceptions are another major building block that may occur during the realization of the process. There are many possible exceptions that may occur: technical exceptions (the message could not be received, or the sequence of the message is incorrect), business exceptions (the message was received but could not be processed, therefore the respective state could not be reached, or the message itself signals an exception, such as Reject Purchase Order) and timeout exceptions (the message did not arrive on time). Business transactions are long running and therefore messages need to be related to each other (in case of acknowledgement or confirmation or a purchase order later on). Data and control dependencies among messages have to be properly implemented in order to support synchronous invocation.

Besides messages, activities, conditions, roles and service providers represent the other building blocks of a service composition. Our framework provides a mechanism to capture these building blocks in the form of classes with special purpose attributes. We use the concept of OOP classes. A building block is defined as an abstract class that contains certain attributes that represent the characteristics of the class. A class along with its attributes represents a business rule. For example, consider a typical scenario of a Trip Planner composite service. A business rule exists on the activity flightBooking that states that the input of this activity should be a message constituting of parts: departureDate; returnDate; from; to and the output from this activity consists of: airline; flightNumber; seatNumber. This rule can be expressed using

an activity abstract class that models a well defined business function.

Activity:

Function= "flightBooking"

Inputs= "departureDate, returnDate, from, to"

Outputs= "airline, flightNumber, seatNumber"

Activities are the operations involved in the composition whereas message abstract class represents a container of information which is used and generated by activities.

Message:

Name= "flightBookingData"

Parts= "departureDate, returnDate, from, to"

Condition class constrains the behavior of the composition by guarding the activities and enforcing pre-conditions, post-conditions and integrity constraints.

Condition:

Name= "departureDateCondition"

Argument= "departureDate"

Predicate= ">"

Value= "currentDate"

The activities participating in the composition are performed by some business roles that are modeled as role class. The instance of this class can be:

Role:

Name= "flightRole"

Type= "airline"

Capabilities= "flightBooking"

Other classes have to be defined also. Flow class defines a block of activities to show how they are connected to each other. It can have attributes (functions, sub-functions, patterns). Partner class describes a party offering concrete services having attributes (name, description, services, cost, quality). These classes are presented to the composition designer in the form of rule templates who can retrieve them and fill the attributes with proper values. Each piece of information pertinent to a class will constitute an instance and is stored in a repository which we call the composition rule repository.

Service Composer

Once the designer has captured all the information the service composer initiates an algorithm which relates the composition rules to construct the composition. The algorithm also follows certain rules to manage the inter-relationships among the classes. Listing 1 shows some of the steps of the algorithm.

1) Determine activities

Against the user request determine the activities with requested functionality.

select activity where activity.function=user.requirement

2) Add message exchange behavior

For each activity involved:

find out the message which contain all the information required for the activity input

select message where

message.parts->includesAll(activity.inputs)

find out the message which contain all the information required for the activity output

select message where

message.parts->includesAll(activity.outputs)

3) Place constraints

For each activity

determine which condition pre-guards the execution of the activity

select condition where

activity.input->exists(condition.argument)

determine which condition post-guards the execution of the activity

select condition where

activity.output->exists(condition.argument)

4) Structure activities

determine the activities that can take part in a flow

select activity where

flow.subfunctions->exists(activity.function)

determine the flows that can take part in a flow

select flow where

flow1.subfunctions->exists(flow2.function)

5) Compose abstract services

For each activity

determine the role interested in carrying out the activity

select role where

role.capabilities->exists(activity.function)

Listing 1. Service Composition Algorithm

The end user could be placing request on a service portal (travel planning service or loan approval service) or the request could be a part of business to business scenario (as described in Section 2). Following the algorithm, a composition is constructed in the form of BPEL file defining the composition and a WSDL file defining the composite service interface.

A mission critical Web services composition must be verified for inconsistencies before making it operational. Unfortunately, detecting process errors at execution time is not only expensive, but also limited to specific use cases. Validation and verification of Web services composition is a separate research area in its own. We shall use techniques described in [13] or [14] to verify the constructed composition. At last, the validated composition is passed on to an orchestration engine which executes it. Several engines are available, for instance, IBM's BPWS4J [8] and Oracle's BPEL Process Manager [9] which are freely available to download.

6. REQUIREMENTS FOR SERVICE COMPOSITION

For the successful creation of Web service composition, the systems should adhere to some basic requirements [12]. These requirements must be addressed when designing business processes involving multiple Web services running over a long period of time. Our proposed framework fulfills all these requirements.

High degree of fault-tolerance: The framework must be able to handle common fault scenarios. There is no guarantee in a SOA that a particular service will be available at a particular time. Additionally, a service interaction may fail due to missing messages or not producing the intended result.

Workflow granularity: The framework should support mechanisms to allow users to generate workflows of varying levels of granularity. For example, abstract and concrete workflows. Abstract workflows specify the workflow without referring to any specific service implementation. A concrete workflow specifies the actual names and network locations of the services participating in the workflow. An abstract workflow allows users to share workflows without reference to any specific service implementation. This is particularly useful as there is no guarantee of the availability of any service in a SOA. On the other hand, a concrete workflow could be useful for provenance purposes. Additionally, workflows of differing levels of granularity should be loosely coupled, i.e. with minimum interdependence.

Specify and refine high-level objectives: The framework should support mechanisms to allow users to specify and dynamically refine a high-level objective which is then translated into a workflow.

User-specified optimization criteria: The framework should provide a mechanism which allows users to specify the workflow composition/execution optimization criteria. For example, the user may want to minimize the total runtime of the workflow or minimize the use of expensive resources.

Scalable: The framework should be scalable to a large number of services. This could be large number of services for the same functionality or a large number of services taking part in the composition process.

Domain independent: The framework should be as generic as possible in order to allow its use within any domain. Regardless of the domain, a composite service represents a service itself which can be invoked by a client and can return results.

Our framework meets these requirements by providing the following:

A comprehensive class model that captures all the requisite information. These classes provide all the necessary data to provide fault tolerance, service bindings, and constraints check.

An algorithm which incrementally composes services from an abstract form to more executable one providing workflow granularity.

Support of refinement of the process by updating the rules.

Domain independence.

7. CONCLUSIONS AND FUTURE WORK

Business to business domain is the most applicable area for Web services composition. Business are always trying to keep up with the growing pace of information technology and business personnel are always trying get independent from the technology underlying their businesses so that they can manage their business processes on their own without the intervention of an IT expert. Our proposed approach provides the business personnel the liberty to design and manage the business process without knowing about the technical details. The most prominent advantage of this framework is the prompt adaptation of the system to the changing environment. Addition or deletion of an activity, change in conditions,

switching among various service providers and any other possible change can be easily incorporated into the system. All what we have to do is to retrieve the template instances and make changes. The system will itself adapt to the change.

The work presented here is at an initial stage. The algorithm and rules described here are stated in a simple way to provide more intuition. In reality composition will likely be much more complex. The composition rules have to be further refined so as to capture more details keeping user involvement as low as possible. A change management sub-system to control the evolution of business rules have to be developed in addition to a prototype to validate our algorithm.

8. REFERENCES

- [1] B. Orriens, J. Yang, and M.P. Papazoglou, "Model Driven Service Composition", *In the 1st Int. Conf. on Service Oriented Computing (ICSOC'03)*, 2003.
- [2] S. R. Ponnekanti and A. Fox, "SWORD: A Developer Toolkit for Web Service Composition", *In Proc. of the 11th Int. World Wide Web Conference*, Honolulu, HI, 2002.
- [3] L. Zeng, D. Flaxer, H. Chang, and J. Jeng, "PLMflow-Dynamic Business Process Composition and Execution by Rule Inference", *Vldb Workshop on Technologies for E-Services (TES02)*, Hong Kong, 2002.
- [4] J. Meng, S. Y. Su, H. Lam, and A. Helal, "Achieving Dynamic Inter-organizational Workflow Management by Integrating Business Processes, Events, and Rules", *In Proc. of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, 2002.
- [5] Wil M.P. van der Aalst, "Web Service Composition Languages: Old Wine in New Bottles?", *In Proc. of the 29th EUROMICRO Conference*, IEEE Computer Society Press, 2003.
- [6] F. Curbera, Y. Golland, J. Klein, F. Leymann, D. Roller, S. Thatte, and S. Weerawarana, "Business Process Execution Language for Web Services", July 31, 2002. <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [7] Business Process Modelling Initiative, "Business Process Modeling Language", June 24, 2002. <http://www.bpmi.org>
- [8] Emerging Technologies Toolkit, "Business Process Web Service for Java", *IBM alphaWorks*, 2003. <http://www.alphaworks.ibm.com/tech/ettk/>
- [9] Oracle Technology Network, "BPEL Process Manager", <http://otn.oracle.com/products/ias/bpel/index.html>
- [10] M. Carman and L. Serafini, "Planning for Web services the hard way", *Workshop on Service Oriented Computing, Int. Symp. on App. and Internet (SAINT-2003)*. Florida, USA, 2003.
- [11] J. Heflin and M.N. Huhns, "The zen of the web", *IEEE Internet Computing*. Vol. 7(5), pp:30-33, 2003.
- [12] S. Majithia, David W. Walker and W. A. Gray, "A Framework for Automated Service Composition in Service-Oriented Architecture", *at 1st European Semantic Web Symposium*, Heraklion, Greece, 2004.
- [13] X. Fu, T. Bultan, and J. Su, "WSAT: A Tool for Formal Analysis of Web Services", *To appear in Proc. of 16th Int. Conf. on Computer Aided Verification*, 2004.
- [14] X. Yi and Krys J. Kochut, "A CP-nets-based Design and Verification Framework for Web Services Composition", *Proc. of 2004 IEEE Int. Conf. on Web Services*, pp. 756-760, San Diego, California, July 2004.

9. AUTHOR CURRICULUM VITAE



Muhammad Adeel Talib is a PhD student at the e-Business Lab, Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan. His work is being supervised by Prof. Dr. Yang ZongKai. (Deputy Head of Electronic and Information Engineering Department, Director of e-Business Lab). He holds a bachelors

degree in engineering and masters degree in computer science and information technology from University of Engineering and Technology Lahore, Pakistan. In recognition of his brilliant educational record he has been awarded Ministry of Science and Technology Scholarship for the year 2000-2002 and Ministry of Education, Cultural Exchange Scholarship for the year 2002-2005. His research interests are e-Business, Web services, distributed systems and Enterprise Application Integration.



Yang ZongKai is a Professor of Telecommunication Engineering and is the Deputy Head of Electronic and Information Engineering Department, Huazhong University of Science and Technology, Wuhan. He is the Director of DSP, Telecomm, e-Business, and e-Learning Research Institutes. He received his bachelor's and master's degree (in the area

of speech recognition) from Huazhong University of Science and Technology (HUST), China, in 1985 and 1988 respectively, and his Ph.D. in 1991 at Xi'an Jiaotong University, China (in the area of Array Signal Processing and Neural Networks). He was sponsored from 1994 to 1995 by KOSEF (Korea Science and Engineering Foundation) to be a post-doctoral candidate to work for the ATM network switch and traffic control in Korea University. In 1998, he was invited by UTStarcom company of America as a visiting professor to work on ATM ring protocol design, which is used for mesh channel establishment of IP Gateways. He is a senior member of Institution of Electronic Engineers China, member of the IEEE. His research interests are in the areas of ATM networks, B-ISDN networks include computer network protocol design by ROOM, access technologies, networks traffic modeling and control, and client/server design for multimedia information services. His other major research activity is in the area of digital signal processing dealing mainly with spectrum estimation, weaker signal detection, neural networks, wavelet, fractal, chaos, adaptive systems and speech processing. In the past 10 years Dr. Yang has published more than 80 journal and conference publications. He is also the editor and co-editor of four academic books (in Chinese).

Link-Based Markov Model Prefetching Algorithm on Web Cache

Wang Zhao, Guo Cheng-cheng , Yan Pu-liu
 School of Electronics Information, Wuhan University
 Wuhan, Hubei, 430079, China
 Email: wangzahao1212@yahoo.com.cn Tel: 13071288242

ABSTRACT

Reducing the web latency is one of the primary concerns of Internet research. Web caching and web prefetching are two effective techniques to reduce latency. However, most previous research has addressed these two techniques separately. In this paper, we put forward a new proxy scheme which integrates web caching and web prefetching. This scheme is based on a site-oriented cache storage structure. Compared with the traditional document-oriented structure, it uses less memory, improves the manageability of local cache storage, and supports intelligent prefetching better. On the basis of this structure, we bring about a Link-based Markov (LBM) prediction algorithm. Our prediction algorithm use access sequences obtained from server logs to create and maintain the 1-order Markov models, then use these models to predict the subsequent possible requests. At last we use trace-driven simulations to prove the feasibility of site-oriented cache storage scheme and test the performance of LBM prediction algorithm.

Keywords: Proxy, Cache, Prefetch, Site-Oriented, Link-Based

1. INTRODUCE

Web browsing dominates today's Internet. More than two-thirds of the traffic on the Internet is generated by the Web^[1]. Much effort has focused on improving the quality of service delivered by the Internet. Caching and prefetching are proven useful techniques for reducing end users experienced latency on the Web.

The fundamental concept of caching is the intermediate storage of popular Web documents close to end users. Caching is effective because many web documents are requested much more than once. While caching alone reduces latency for previously requested documents, web documents prefetching could mask latency for previously unseen, but correctly predicted requests. Prefetching has been used to great advantage in file system, and researchers have already proposed several methods of web prefetching, such as Wcol^[2], PPM^[3], Top-10^[4], etc. However, their studies only focused on an unrealistically small size prefetching buffer, and did not address the interaction between web caching and web prefetching.

Web caching and prefetching can be deployed at various positions in the Internet. Our study focuses on web proxy servers. A proxy server is usually located at the edge of a LAN, intercepting HTTP requests and responses between clients and web servers. An advantage of web proxy caching and prefetching is that all clients within the LAN can share objects stored in the cache.

In this paper, we put forward an approach to integrating web caching and web prefetching by using link correlation of the documents in cache to create the prediction model. The rest of this paper is organized as follows. In section 2, we briefly introduce a site-oriented cache scheme including the cache storage structure and the site-based LRU replacement algorithm. In section 3, we present a link-based Markov predictive model. The implementation of the integrated proxy server is described in section 4. Some experimental results are presented in section 5. In section 6, we present our conclusions and directions for future study.

2. SITE-ORIENTED CACHE SCHEME

Caching is a mature technology widely used in many areas such as operating systems and database systems. Currently, the WWW becomes another popular area to apply caching.

The traditional web cache storage schemes are document-oriented, for example Squid^[5]. They treat all the cache documents equally, although these documents are different in kind, size, and come from different sites. Those documents come from the same site cannot be distinguished from the others. So the useful information between them, such as links correlation, cannot be obtained. It is hard to implement intelligent prefetching based on such cache storage schemes. In order to add intelligent prefetching to our proxy server, we introduce a site-oriented cache storage scheme.

2.1 Cache Storage Structure

After considering those factors such as efficiency of utilizing cache storage, efficiency of locating cache documents, manageability, etc, we decide to adopt the cache storage structure, as shown in Figure 1, in our site-oriented cache storage scheme.

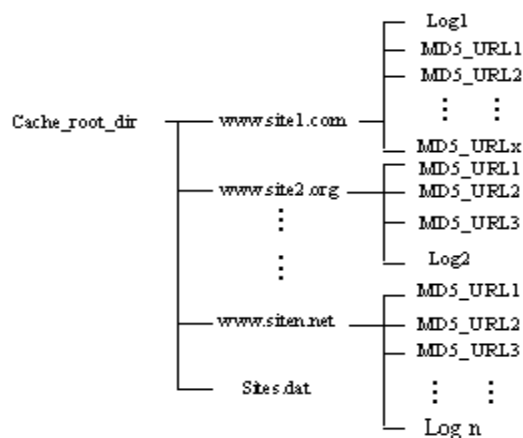


Figure 1 Site-Oriented Cache Storage Structure

This storage structure manages cache documents according to which site they come from. In the cache root directory, those documents coming from the same site are placed in the same subdirectory, and the site name is used to name this subdirectory. Each document in cache is named by the string generated by MD5 operation using its URL. There has a log file named 'sites.dat' in the cache root directory, which logs the number of sites, visiting frequency and average visiting delay of each site, etc. In each site subdirectory, there also has a log file which records the sizes, visiting time, visiting date, visiting sequence, link correlations of cache documents in this subdirectory. These log files can be used by our cache replacement algorithm and prediction algorithm.

2.2 Cache Replacement Algorithm

According to the site-oriented cache storage structure, our cache scheme ameliorates the traditional cache replacement algorithm.

We use real-time replacement algorithm based on site. As the amount of cache storage is as huge as hundreds of GB nowadays, the performance discrimination of those traditional replacement algorithms is negligible. So we use the LRU as model, design the site-based LRU replacement algorithm. We use a linked list to keep and sort the sites in the cache. When the cache storage is too small to accommodate new documents, the rear-end sites of this list will be removed. That means the subdirectories and all the documents in them will be removed, till the cache storage is big enough to accommodate the new comer. Our replacement algorithm has two advantages. Firstly the linked list we used is based on site, so it is much smaller than the one based on document, and uses the less memory. Secondly we will remove all the documents in a subdirectory at one time, so more storage space is vacated for the new comers and the times of call replacement is decreased.

Besides the real-time replacement based on site, we use a routine maintenance algorithm based on documents, acted as the supplement for the real-time replacement algorithm. For the real-time replacement algorithm is based on site, it ignores the difference between the documents in the same site subdirectory. There may be cool documents in a hot site or hot documents in a cool site, our real-time replacement algorithm is not fit for these two cases. So we do a routine job to clean up the whole cache storage based on documents. We use one of the traditional replacement algorithms as GD-SIZE to do this job.

3. LBM PREDICTION ALGORITHM

The next page that users would probably request is related to the current page, and has no relation with the pages they have browsed before. That means if the current state is confirmed, the future state has no relation with the past states. This is Markov's characteristic. As the behavior of visiting WWW accords with Markov's characteristic, it is appropriate to use Markov model to analyze it. Actually, the famous PPM prediction algorithm utilizes the high-order Markov model^[3,6].

Based on the site-oriented cache storage scheme described in section 2, we introduce a new link-based Markov (LBM) prediction algorithm. Our algorithm will create a 1-order Markov model for each site in cache. When a new request

comes, our algorithm will prefetch documents for this request based on the corresponding LBM model. If new request call a new site, that means there has no corresponding LBM model, our algorithm will prefetch the hyperlinks according to our own rules.

3.1 Create The LBM Model

Traditional Markov prediction algorithms, such as PPM, classify the requests only by temporal sequence, but it is not enough for WWW applications. For example, users can open many links in a new window, and they also can open many windows to browse different sites. For another example, proxy will receive several requests in sequence. Though these request may be come from the same IP, or they may be ask to the same Web server, they probably come from different users. Thus, we cannot deduce states-transfer by the traditional classifying method. So, the more exactly we want to predict next request, the more exactly we should classify the requests.

When classifying requests from our proxy log file, we use such algorithm as below:

- 1) Clean the requests asking for pictures, because most of these requests are automatically sent by browser, not by users.
- 2) Dissect the log file into different sections by destination IP, the requests asking for different destination IP must belong to the different access sequences.
- 3) Dissect the sections into smaller sections by source IP, the requests come from different source IP must belong to different access sequences.
- 4) Dissect the small sections further. In each small section, if the interval between two requests is more than a threshold, for example 10 minutes, named time-window, we decide they belong to different access sequences.
- 5) Moreover, in each small section, if the two requests coming in sequence cannot link to each other directly, we decide they belong to different access sequences.

We get a set of access sequences by the upper algorithm. This set can be used to create and train Markov states-transfer tree. 0-order Markov model directly use the page visiting probability, $p(x_i)$. 1-order Markov model use the conditional transferring probability of page to page, $p(x_i|x_j)$. k-order Markov model use the conditional transferring probability of the access sequences to page, $p(x_n|x_{n-1}, x_{n-2}, \dots, x_{n-k})$. Thus, we can get the tree of past requests states-transfer, which can be used to predict the next request. In general, the higher order the Markov model has, the more exactly it predicts. But the high-order Markov model need store a mass of state, thus consumes much memory. Moreover it will consume a lot of time to compute, which damages the real-time performance. For the reasons above, it is not fit for our prediction algorithm to use high-order Markov model. So we adopt 1-order Markov model, as shown in Figure 2. For every document in cache, we use a visiting counter and a set of link-transfer counters to describe it. The visiting counter is used to record the visited times of this document, that is C_A in Figure 2. The number of link-transfer counters is equal to the number of hyperlinks in this document, and one of link-transfer counters is used to record the visited times of one of

hyperlinks in this document. That is $C(A, B_i)$ in Figure 2, $i=1\sim k$, and k is the number of hyperlinks being visited in document A. So the conditional transferring probability of page to page is:

$$p(B_i | A) = C(A, B_i) / C_A \quad (1)$$

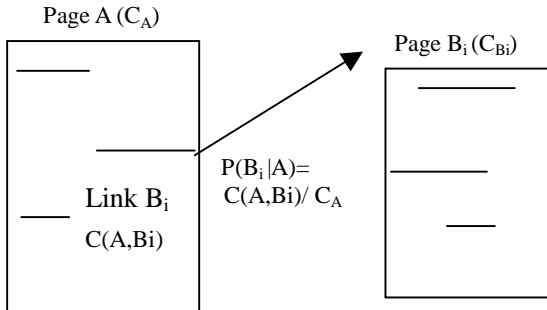


Figure 2 1-order Markov Model of Web Document

We use a reticulate linked list to describe this model (see Figure 3). Every node represents a document logged in the log file, and every directional line represents a hyperlink. Because we have used site-oriented cache storage scheme in our system and dissected the log file according to destination IP, so we can ignore the link correlation between the different sites and maintain a individual states-transfer map for each site.

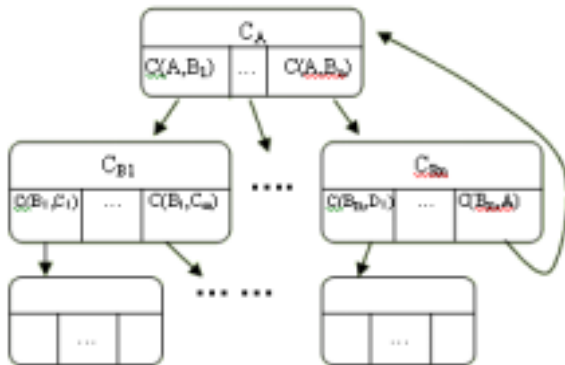


Figure 3 LBM Model of Web Site

The pseudocode of our algorithm to create the LBM model from the corresponding access sequence is described as following:

```

Input :   Historical Web access sequence database
Output :  Link-based Markov Model
Model_Creation (Historical Web access sequence database)
{
  For Each sequence  $i$  do
    Current = Find_Node(1st event)
    If Current = NULL Then
      Current = Create_New_Node(1st event)
    EndIf
    Count of Current node ++
    For Each Event  $j$  of sequence  $i$  do
      If URL of Current's child node = URL of Event  $j$ 
      Then
        Count of this child node ++
        Current_Child = this child node
      
```

```

    Else
      Current_Child = Create_Child_Node (
        Current, event  $j$  )
    EndIf
    Count of this link ++
    Count of Current_Child node ++
    Current = Current_Child
  EndFor
EndFor
}

```

After having collected enough access log, our system will analyze the log and attain the set of access sequences, using the set to create original LBM models.

Web sites change their contents from time to time, furthermore clients often change their visiting habits. So our site models cannot keep unchanged after created, and we should update them often. One method is to update the model when complete the service for a new request, which is named real-time update. Considering the efficiency of our system, we decide to update the LBM models periodically in the fixed time, rather than update them on-line when needed.

3.2 Prefetch Pages From New Sites

For our LBM prediction algorithm is based on historical access data, it cannot work when clients request a web page in a new site.

The Top-10 algorithm can prefetch in this case. It needs the destined web server provides statistical data^[4], but not all types of servers support it. The other algorithm is parsing the content of the web page. But this algorithm is difficult to implement and will introduce complexity into our system. We discover these facts when analyze the users' visiting behavior:

- 1) When the words describing a hyperlink include some frequently used keywords, users will probably hit this hyperlink. These keywords include 'next', 'more', 'new', 'hot', and so on.
- 2) When the words describing a hyperlink include the same topic words with header of the current web page, users will probably hit this hyperlink.

So we create a dictionary including some frequently used keywords and another topic dictionary by hand in advance. The administrator can add to and delete from these two dictionaries. When meeting a page from a new site, our system will simply scan all of the words describing hyperlinks. If any word is included in the keyword dictionary, our system will prefetch the hyperlink described by this word. If no word is in the keyword dictionary, we will scan the topic dictionary. If still none, we will prefetch the first n hyperlinks in order, just like Wcol algorithm does.

In sum, our real-time prediction algorithm is described as below:

```

Input :   Current Page, Link-based Markov Model
Output :  Predicted URLs
LBM_prediction(Current Page, LBM Model)
{
  Current = Find_Node(Current Page)
  If Current != NULL Then

```



```

    Prediction = URLs of Current's child nodes
  EndIf
  If Prediction = NULL Then
    Prediction = Find_Key_Words(Current Page)
  EndIf
  If Prediction = NULL Then
    Prediction = Get_First_N_Links(Current Page)
  EndIf
}

```

4. IMPLEMENTATION

Our server is comprised of three modules, which are proxy module, caching module, and prefetching module. These three modules cooperate with each other. The proxy module processes the TCP links with clients and web servers, parses and forwards HTTP requests and answers. The caching module manages the local cache storage including search, replacement and checking cache consistency. The prefetching module creates and maintains LBM models for cache sites, predicts the next pages and prefetchs them, controls the condition of prefetching.

Our system run at a PC of Intel Pentium III 550 , 128M SRAM , 10G Hard-disk, using Red Hat Linux 9.0, acting as the gateway servicing for our workgroup.

5. EXPERIMENTS AND RESULTS

We use the access log of our proxy server to do trace-driven simulations experiment to demonstrate the benefits of our intelligent proxy. The log was collected from 09:00, July 10 2003 to 09:00, August 8 2003, a total of 30 days.

5.1 Feasibility of Site-Based LRU Replacement

We compare our replacement algorithm with traditional LRU algorithm in the cache hit ratio, the result can be see in figure 4. When we calculate hit ratio, we only concern those GET requests which meet a cache-hit answer in the traces and ignore the cache consistency. Set cache size to 1G.

Use the following Eq2 to calculate cache hit ratio:

$$HR = N_h / (N_h + N_m) \quad (2)$$

N_m is the number of cache-miss requests, and N_h is the number of cache-hit requests.

We use these two algorithms to calculate cache hit ratio, then plot the points to two lines.

From Figure 4, it can be seen that LRU algorithm is not better than our site-based LRU in performance of hit ratio, which is most important performance to cache server. Moreover, this result is obtained in a small cache size. As we have metioned in section 2, the discrimination of hit ratio between different replacement algorithms is negligible when cache storage is large scale. So, we can conclude our site-based LRU replacement algorithm is feasible.

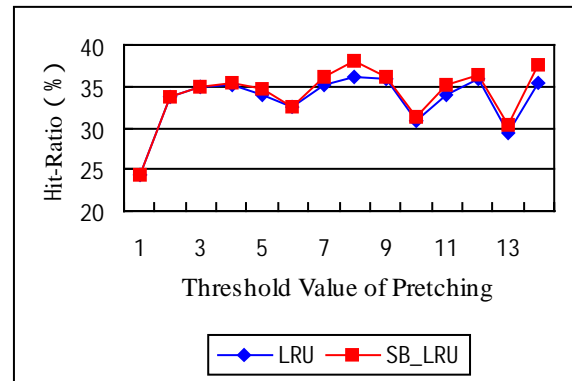


Figure 4 Comparison of LRU and Site-Based LRU in Hit Ratio

5.2 Performance of LBM Model Prediction

We compare among Wcol, 1-order PPM and our LBM model algorithms in prefetching hit ratio, see Figure 5. Set the time-window used to dissect log file to 30 minutes, and the time-window used to calculate prefetching hit ratio to 2 minutes.

Use the following Eq3 to calculate cache hit ratio:

$$Prefetch - HR = \frac{C_{accurate}}{C_{all}} = \frac{C_{accurate}}{C_{accurate} + C_{wrong}} \quad (3)$$

$C_{accurate}$ is the number of web documents which are requested after they have been prefetchd. C_{all} is the whole time of predictions. Prefetch-HR represents veracity of predictions.

We use three algorithms to calculate prefetching hit ratio at different prefetching thresholds, then plot the points to three lines.

From Figure 5, it can be seen that our LBM model prediction algorithm predicts more exactly than Wcol and PPM, especially when the low prefetching threshold. We can conclude that our LBM model prediction algorithm is useful to enhance the prediction veracity.

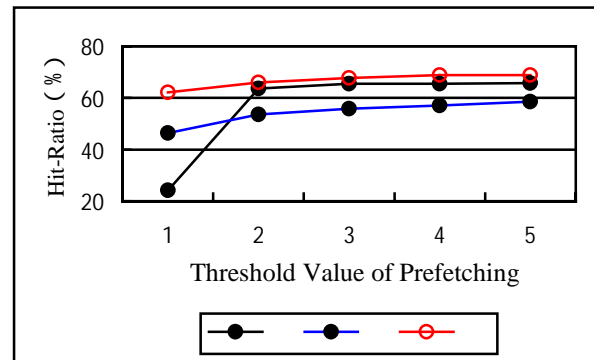


Figure 5 Comparison Among Three Prediction Algorithms in Hit Ratio

represents Wcol algorithm

represents PPM algorithm

represents our LBM model prediction algorithm.

6. CONCLUSION AND FUTURE WORKS

In this paper, we present an approach to combine web caching and web prefetching in the proxy server applications. We propose a site-based caching storage structure to store the web documents according to which site it comes from. We also propose a link-based Markov model prediction algorithm to prefetch the probably visited web documents for clients according to the corresponding Markov model. By using trace-driven simulations, we show that our system integrated caching and prefetching could improve the manageability of caching storage and veracity of prediction.

In our future work, we will try to use more sophisticated prediction model to see how effective the integrated system is, for example using the high-order Markov model. We will also try to use our system in a cluster environment.

7. REFERENCES

- [1]. G. Barish and K. Obraczka, World Wide Web Caching: Trends and Techniques, IEEE Communications Magazine, May, 2000, p178-185.
- [2]. Wcol Group, WWW collector: the prefetching proxy server for WWW, 1997, <http://shika.aist-nara.ac.jp/products/wcol/>.
- [3]. Xin Chen and Xiaodong Zhang, Popularity-Based PPM: An Effective Web Prefetching Technique for High Accuracy and Low Storage, Proceedings of the International Conference on Parallel Processing, 2002, p296-304.
- [4]. E. P. Markatos and C. E. Chronaki, A Top-10 Approach to Prefetching on the Web, ICSFORTH Technical Report, Aug, 1996, No.173.
- [5]. Squid Programmers Guide. <http://www.squid-cache.org/Doc/Prog-Guide/>.
- [6]. Xing Dongshan and Shen Junyi, A New Markov Model for Web Access Prediction, IEEE Computing in Science and Engineering, Nov, 2002, p34-39.



Wang Zhao is a graduate student of Network Communication Lab in School of Electronics Information, Wuhan University. His major is Communication and Information System. He graduated from Wuhan University in 2001 with specialty of Communication Engineering. He has published two Journal papers. His research

interests are in Web caching, Web prefetching and cluster technology.

Analysis and Comparison between Two Distributed Object Technologies CORBA and DCOM

He keyou, Zhang weilin

Department of Computer Science, Wuhan University of Technology,
Wuhan, Hubei Province, China

Email: zhangweilin_aq@yahoo.com.cn Tel: 027-86555531

ABSTRACT

Distributed Object Computing is the development direction of distributed computing, CORBA and DCOM are two most important distributed object computing platforms. This paper made a deep research on the structure and characteristic of the two technologies, and summarized their advantages and disadvantages. It also compared the aspects of the structure and regulatory, the ability of astride-platform and integrating different language, the problem of security, invoking method, the degree of separation between Client and Server, the communication protocol and the multi-thread. Hence, the two distributed object technologies CORBA and DCOM can be fully understood and their applications can be grasped through comparative study of their similarities and differences. As a whole, the performance of CORBA is better than DCOM, but DCOM has its advantages, hence, DCOM will exist with CORBA for a long time, and they will cooperate even tightly in the future.

Keywords: Distributed object computing, Object- Request Broker, COM, CORBA, DCOM

1. INTRODUCTION

The structure of Client/Server is one of noticeable technologies, it has become basic computing structure which substitutes with the system structure of "host-terminal". At first, Client/Server adopted the Client/Server technology which SOCKET conveys data. Now, as for the ripe network technology and the development of OOP, the study and application climax of distributed object technology are carried out and the structure of Client/Server is also in the transition from conveying "data" to conveying "data" and "program" (namely "object:") at the same time. Letting standard software packages distribute in the network and call each other is the solution which enterprise expects. But creating a system which can let all software communicate transparently and use the other side service each other, no matter these objects are in the same arranging address space or in the different arranging address space or in the different computer, is the challenge oriented-object distributed computing is facing. If no methods let object call each other in the network, the application will be constrained. About distributed computing based on object there are two kinds of famous technologies: OMG'S CORBA(Common Object Request Broker Architecture) and Microsoft DCOM(Distributed Component Object Model), the paper will analysis and compare the two technologies.

2. THE SUMMARY OF CORBA AND DCOM

2.1 CORBA

CORBA is a standard oriented-object application program specification, it is made by OMG organization. It is also, CORBA is a method to solve the inter-connection of hardware and software in DCE; CORBA is composed by the following components.

(1) Object Request Broker (ORB)

It makes object create and receive request and answer transparently in the distributed environment, it is the foundation of distributed object application and it is also the foundation of interaction among applications in the isomorphic and isomeric environment, it stipulates the define of distributed object (interface) and language map and realizes the communication and mutual operation. ORB can regard as a middle component of building Client/Server relation among objects. Based on ORB, Client can call the method which service object supplies, the service object can run in the same computer with Client, it also can run on the other computer and realize mutual operation with Client through network. ORB intercepts the request which client sends and is responsible for finding the service object which realizes the request in the software bus, then finishes parameter and method calling and returns the final outcome.

(2) Object Service

It is a service set to support the basic function of using and realizing object. It is necessary to create every distributed application and often independent to application fields. Object Services Specification is included in Common Object Services Specification of CORBA.

(3) Common Facilities

It is a service set. Many applications share the set. For example, the system management or the e-mail equipment belongs to common facilities.

(4) Application Objects

It is a kind of products to control the interface, it is supplied by individual supplier. It obeys traditional application concepts, hence, it isn't been standardized by OMG. Per contra, it constitutes the highest lever of relational model.

(5) Domain Facilities

It supplies the services object which is intimate relevant to application fields, it supports the study in many fields, such as telecommunication and finance, etc.

2.2 DCOM

DCOM is the distributed computing strategy, it is adopted by Microsoft and DEC, etc. DCOM offered by Microsoft in 1996, it is based on the following thoughts: different application programs can take mutual operation with DCOM protocol. COM has experienced from OLE2/COM, ActiveX, DCOM to COM+. At present, COM+ joins MSMQ and MTS, it is a relatively perfect platform for distributed object computing. DCOM is based on COM (Component Object

Model), and COM is the foundation of OLE and ActiveX and has become the indispensable Windows component. COM specifications can regard as a kind of object level structure, such as MFC or VCL, but the difference is that COM specification is independent to language among COM, MFC and DCOM, so there are many tools to create COM and DCOM components, such as Visual C++, RAD (Rapid Application Development) etc, at the same time there are a large number of ActiveX components which have been built and can be used directly. Although DCOM is suitable to Windows environment at first, it is expanding to UNIX (such as Solaris etc) platform. Although DCOM adopts the technology of oriented-object, it doesn't pass directly, but realize remote services by remote procedure call (RPC). The flow of work for DCOM is the following:

Step 1: Client Initialization (Client Side)
 Step 2: Server Activation (Server Side)
 Step 3: Call Class Factory
 Step 4: Multiple Query Interface (MQI)
 Step 5: Proxy/Stub Loading
 Step 6: Method Call

As for DCOM develops from COM, all application programs, components, tools and knowledge based on COM can use directly. DCOM has the following characteristics:

- scale variable
- supply abundant, balanced communication among components.
- expand new function easily.
- own a large number of available components.
- utilize network band width effectively, supply good response for end user.
- inherent security.
- automatic loading balance and the ability of fault-tolerant.
- take effective configuration and management easily.
- support every network protocols.
- use TCP/IP protocol.

3. THE SAME ADVANTAGE AND DISADVANTAGE

CORBA and DCOM inherit and develop the thought of PRC (Procedure Remote Call), they will encapsulate the interactive operation among distributed application programs with the semantics which the local procedure calls. They both make Client application program call remote objects as the local objects, having good transparency and performance. As programmer, there isn't much difference between programming for distributed application program and for the local program, they needn't learn and use complex network API, it is enough for programmer to have the method of programming for the local application program. They can put their energy emphasis on the design of interface and algorithm. At the same time, CORBA and DCOM adopt the thought of oriented-object. Compared the method of oriented-object with the method of traditional structure, the former is much better than the latter in the software analysis, design, encoding, maintenance. The primary element of remote access interface among distributed programs is object, at fact remote call is that calling the method of these objects. CORBA and DCOM support the thought of encapsulate and polymorphism of oriented-object. Moreover, CORBA supports inheritance, but DCOM don't support inheritance, so generally speaking, DCOM doesn't adopt all the

characteristic of oriented-object. They have the same targets: the targets of distributed object technology are seamless connection among object components and plug&play. One of basic conceptions for distributed object technology is the conception of component, component can astride platform, network, language, application program, tool, hardware. Distributed object component will change the traditional mode thoroughly which product software. Although CORBA and DCOM both adopt the method of oriented-object and calling remote object as local object, they only quote remote object. Transferring remote object to Client terminal is the important function of distributed software, but DCOM and every version before CORBA 3.0 don't solve the problem. CORBA 3.0 specification adopts the method of transferring object through value, but it can't solve the problem thoroughly.

4. THE DIFFERENCE

We compared CORBA with DCOM from the following aspects:

(1) The Structure and Specification

CORBA is a general distributed object specification, it doesn't give the referenced realized program, so it is very flexible. But DCOM has explicit realized background, specification, so it isn't benefit to optimize.

(2) The Ability of Astride-Platform

CORBA isn't presented by manufacturers, but by standard organization, so it is independent platform at first. DCOM is presented by Microsoft at first, it is confined to Win32 platform, then Software AG will expand it to other platforms, but the performance of DCOM in other platforms isn't good.

(3) The Security

All distributed computing includes communication. If distributed computing is in the distributed network, the security and the integrity of data will face the challenge while transferring data. The security must assure user won't be damaged by destructive code. DCOM communicates among objects in different place by RPC. It isn't supplied the safe assurance in the distributed data network (such as Internet). ActiveX controls which realizes with DCOM don't include strict security check or resource authority check, controls has all authorities of resources, so inherent security is lack. But OMG has specified the security service for the system based on CORBA, the service not only supplies the security and authentication, but also realizes non-denial. The most services of CORBA are defined by OMG, but the transmission and security services are based on the standard of DCE. The communication regime among objects in CORBA is based on the message delivery of ORB, but in DCOM is ORPC, it is the extension of DCE RPC, adding a kind of new data type, namely, object quoted type.

(4) The Ability of Astride-Language

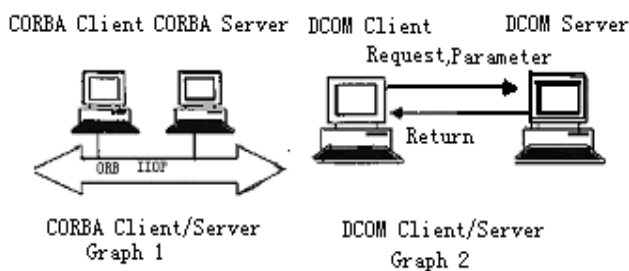
The language is almost c++ in the realization of DCOM, but the support of other program language is difficult. CORBA has the characteristic of language independence, so it can support many languages. OMG has adopted the specification which uses CORBASM regime with c, c++, ADA and Smalltalk.

(5) The Call Mode

CORBA permits client application programs to call the services objects in two ways, static and dynamic. Client should point which services component and which method is called in the static way, so client programs depend on the services object, at fact, the way is not the essence of CORBA. In the dynamic way, the interface of services object is unknown at first, client terminal finds the services object through Naming Server or Trader Server provided by CORBA while running, then calls the services method by DII, so client program can use services provided by objects which is unknown while writing, this is true advantage of CORBA. Compared with static call, dynamic call is more flexible, it can supply new service and method at any time. On the other hands, programming for dynamic call is more complex, the execution time is longer. As same as CORBA, DCOM also supports two object calls, static and dynamic, but at fact there is little difference. No matter which call way is, their implements both depend on the interface which IDL describes. But DCOM supplies two interfaces, different call way needs different interface. A interface and its components should be defined in the static call way, let MIDL compiling program create automatically proxy-stub code which connects these components, but DCOM supports dynamic call by class library. Class library includes these files which describe components, client application obtains these interface through COM interface called IDispatch, the advantage of which is ,no need to define the interface by hand, letting MIDL create proxy-stub code, using default IDispatch proxy-stub code.

(6) The Degree of Separation between Client and Server

Being CORBA adopts Broker, the function of Broker is: finishing the map which gives service request to client terminal, automatically finding server object, then executing in the server. Client programs don't link directly with server, but be only interactive with ORB, it is also, Client and Server separate completely (Graph 1). DCOM Client programs call DCOM object method by interface point, moreover, at first instanced DCOM class before using its interface, so Client and Server link directly (Graph 2).



(7) The Network Communication Protocol

The communication between CORBA and ORB or ORB and remote object complete through communication protocol IIOP (Internet Inter Orb Protocol), IIOP belongs to TCP/IP specification, and the communication protocol between WWW server and explore program will probably change from HTTP to IIOP. But DCOM adopts UDP to reduce communication overland and delay.

(8) The Multi-Thread

CORBA supports the multi-thread, it is also, permits multiple Client program to call the CORBA service object at the same time. Server creates a thread for every Client connection to handle multiple Client request at the meantime.

But global memory variable and data should be protected while programming. But DCOM supports Apartment Model multi-thread mode, it is also, process can be made up of multi-thread, but DCOM object can't be multi-thread. Every DCOM object only runs in single thread.

(9) Others

All interfaces in IDL of DCOM inherit from a common class, IUnknown, but there isn't common super class in CORBA. DCOM supplies flexible running binary standard for distributed object system, but the emphasis of CORBA is static system structure. CORBA supplies abundant abstract, encapsulated regime, but DCOM obtains more flexible running environment at the sacrifice of the aspects. The method in CORBA permits to return every legitimate type, but in DCOM the method only permits to return the 32 bits result.

5. CONCLUSION

CORBA and DCOM are both important platforms in distributed computing system, the paper summarized their advantage and disadvantage and compared their performance. As their individual characteristic, they will be applied to different aspects. DCOM should be the best choice for Microsoft technology system, but as far as astride-platform is concerned, CORBA should be the best choice. But with technology constant harmony, the absoluteness of choice will shrink gradually.

6. REFERENCES

- [1] Keahey K, Gannon D. PARDIS: A Parallel Approach to CORBA [R]. Technical Report IUCS TR 475.Indiana University, February 1997, pp 131-140.
- [2] The Architecture of DCOM [Z]. Microsoft Whitepaper, 1996.
- [3] The Technical Overview of DCOM [Z]. Microsoft Whitepaper, 1996.
- [4] Wang Bo, Wang Hongman, Zou Hua, the Distributed Computing Environment [M], Beijing: Beijing Posts& Telecommunications Press. 2000.
- [5] Microsoft corporation, The Architecture of DCOM[EB/OL]



He Keyou was born in March,1956.He is a Vice Professor in Wuhan University of Technology, he graduated in Wuhan University of Technology.His research interests are in informa-system and database system.

Zhang Weilin : was born in 1980.He is a postgraduate in Wuhan University of Technology. His major is Database technology.