2001International Symposium on
Distributed Computing and Applications to
Business, Engineering and Science

DCABES 2001

PROCEEDINGS

Editor in Chief Guo Qingping

Hubei Science and Technology Press, Wuhan, China

2001 International Symposium on
Distributed Computing and Applications to
Business, Engineering and Science

DCABES 2001

PROCEEDINGS

Editor in Chief Guo Qingping

Hubei Science and Technology Press, Wuhan, China

图书在版编目(CIP)数据

2001年电子商务、工程暨科学领域分布式计算和应用论文集==DCABES 2001 Proceedings/郭庆平主编.
武汉:湖北科技出版社,2001.10
ISBN 7-5352-2722-8

Ⅰ.2… Ⅱ.郭… Ⅲ.分布式处理系统-国际学术会议文集-英文 Ⅳ.TP316.4-53

中国版本图书馆CIP数据核字(2001)第071549号

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
MOE Ministry of Education, China
NSFC National Nature Science Foundation of China
IBM DeveloperWorks

NSFC

IBM

DCABES 2001 PROCEEDINGS © Editor in Chief Guo Qingping

Editorial production by Li Yueping Cover art production by COLORPRINCE

Published by ***Hubei Science and Technology Press, Wuhan, China*** Tel: +86-(0)27-86782508
Address: 75 Huangli Road, Wuchang, Wuhan, China Post Code: 430077

Printed in Wuhan, China by Youngster Union Printery Post Code: 430063

787mm×1092mm sixteenmo 17.25 sheets 1 300 k Characters
First Edition October 2001 First Impression October 2001

ISBN 7-5352-2722-8/TP.64 Price RMB 280

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

Contents

Parallel/Distributed Algorithms

Image Processing and Multimedia Applications

Parallel/Distributed Computational Methods in Engineering

System Architectures, Networking and Protocols

Web-Based Computing & E-Business

Network Security

Poster Presentations

Foreword

With the dramatic development of computer and communication technologies, distributed computing and applications play an increasingly important role in many areas, such as electronic commerce, engineering design, scientific computing and manufacturing. Many research results and new ideas in those areas emerge from time to time and the distributed computing makes more and more effects to the people's daily lives of our global village.

The 2001 International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES 2001) is a major international gathering in year 2001 in China. It provides a forum for HPC researches, designers and users throughout the world to exchange their ideas, case studies and research results related to the issues of high performance computing based on distributed systems. We are fortunate to have world-renowned experts to give insightful keynote speeches at the conference. The reports and papers collected in the Proceedings of DCABES 2001 cover many fields of distributed computing and will be beneficial to people working on those areas. I am sure that you will find the DCABES 2001 both enjoyable and productive.

It is with this perspective that we are very pleased to have the DCABES 2001 held in Wuhan and to receive colleagues from dozens of countries over the world. We are grateful to the China Ministry of Science and Technology (MOST), the China Ministry of Education (MOE), the Natural Science Foundation of China (NSFC). It is their supports that make the DCABES 2001 held successfully here. We also would like to thank the Wuhan University of Technology, China (WUT), the International Science and Technology Cooperation of Hubei Province, China (ISTCA), and the Computer Academic Association of Hubei Province & Wuhan Metropolis, China (CAA) for their efforts made as the local organizers of the conference.

I would like to take this opportunity to thank all people in the scientific committee, the organizer committee and the editorial board of the DCABES 2001 proceedings, who have helped produce a successful conference programme. We must pay special tributes to Dr. Lai Choi-Hong, who is the driven force behind the DCABES 2001, Prof. Guó Qingping together with the programme committee and the external reviewers, who devoted many efforts in handling the paper reviewing process.

We wish you will all enjoy your stay at the conference, and have a wonderful time in Wuhan, China.

Lu, Professor Xicheng
NUDT, China
Fellow of CAE, China
Honorary Chair of the DCABES 2001

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 new millennium, 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 order to reflect this trend, on the last August in a seminar held in Hong Kong Polytechnic University, some members of the scientific committee of the DCABES 2001 proposed an international conference concentrating on the distributed computing and its applications on business, engineering and sciences. It was agreed that the DCABES 2001 would be held in Wuhan China. We are gratified that this idea is well received by our colleagues all over the world, who responded by submitting papers covering a wide range of topics, such as Parallel/Distributed Algorithms, Image Processing and Multimedia Applications, Parallel/Distributed Computational Methods in Engineering, System Architectures, Networking and Protocols, Web-Based Computing & E-Business, Network Security and various types of applications.

Papers submitting to the conference come from over 12 countries and regions. All papers contained in this Proceeding are peer-reviewed and carefully chosen by members of Scientific Committee, proceeding editorial board 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 proceedings editorial board and external reviewers for selecting the 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. I am also grateful to Prof Chris R. JESSHOPE, Prof. Lishan Kang, Prof. Nengchao WANG, Prof. Yakup Paker as well as Dr. Choi-Hong LAI, for their contributions of keynote speeches in the conference.

Sincerely thanks should be forwarded to the China Ministry of Science and Technology (MOST), the China Ministry of Education (MOE), the Natural Science Foundation of China (NSFC). Without their supports the DCABES 2001 could not be held in Wuhan China successfully. We would also like to thank the WUT (Wuhan University of Technology, China), 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 IBM DeveloperWorks (www.ibm.com/developerWorks/cn) made some contribution to the conference.

Finally I should also thank A/Professor Jian Guo and A/Professor Xianqiao Chen for their efforts in conference organizing activities, my postgraduate students, such as Mr. Jing Gong for the conference website design, Mr. Dun Mao for his efforts in organizing activities and other colleagues and students such as Mr. Pingbo Xiang, Mr. Hongjun You, Mr. ChunMing Yuan, Ms WenPing Zhang, Mr. Lei Yang, Mr. ChunShui Liu, Mr. JinHua Liu, for their time and help. Without their time and efforts this conference cannot be organized smoothly.

Enjoy your stay in Wuhan. Hope to meet you again at the DCABES 2002.

Guo, Professor Qingping
Chair of the DCABES2001
Dept. of Computer Science
Wuhan University of Technology
Wuhan, China

Honorary Chair
Lu, Professor Xicheng, NUDT, China, Fellow of CAE

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

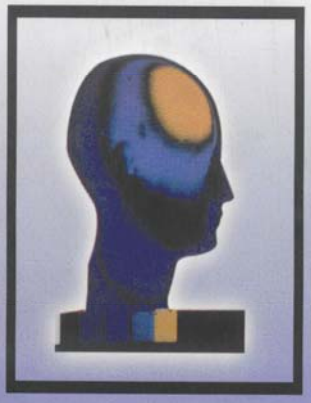
Guo, Professosr Q. P., Wuhan University of Technology
Ho, Dr. P. T., University of Hong Kong
Kwan, Mr. W. K., University of Hong Kong
Lai, Dr. Choi-Hong, University of Greenwich
Lee, Dr. John, Hong Kong Polytechnic University
Liddell, Professor Heather, Queen Mary and Westfield College, University of London
Loo, Dr. Alfred, Lingnan University
Lu, Professor Xicheng, National University of Defense Technology, China
Ng, Dr. Michael, University of Hong Kong
Yakup, Professor Paker, Computer Science Department, London University
Tsui, Mr Y M Thomas, Chinese University of Hong Kong

Local Organizing Committee

Chen, A/Professor Xianqiao. Wuhan University of Technology, China
Guo, A/ Professor Jian. Wuhan University of Technology, China
Guo, Professor Q. P., Wuhan University of Technology, China
Kang, Professor Lishan, Wuhan University, China
Lu. Professor Zhengding, Huazhong University of Science and Technology, China
Wang, Professor Nengchao, Huazhong University of Science and Technology, China
Xiong, Professor Qianxing. Wuhan University of Technology, China
Zeng, Professor Chunnian, Wuhan University of Technology, China
Zhou, Professor Zaojian. Wuhan University of Technology, China
Zhu, A/Professor Dengyan. Wuhan University of Technology, China

Editorial Board

Cover art production by COLORPRINCE



ISBN 7-5352-2722-8
9 787535 227225 >

ISBN 7-5352-2722-8
Price RMB 280/YUAN

Asynchronous Parallel Evolutionary Algorithms for Optimizations

Lishan Kang, Yan Li, Zhuo Kang, Pu Liu, Yuping Chen
State Key Laboratory of Parallel and Distributed Processing,
State Key Laboratory of Software Engineering,
Wuhan University, Wuhan, 430072, China
E-mail: kang@whu.edu.cn

Hugo de Garis
Starlab, Brain Builder Group, Brussels, Belgium, Europe.
E-mail: degaris@starlab.net

ABSTRACT

Recently we proposed a Robust Evolutionary Algorithm for solving nonlinear programming (NLP) problems [4]. It is an extension of Guo's algorithm [1], which possesses enhanced capabilities for solving NLP problems. These capabilities include: a) advancing the variable subspace, b) adding a search process over subspaces and normalized constraints, c) using an adaptive penalty function, and d) adding the ability to deal with integer NLP problems, 0-1 NLP problems, and mixed-integer NLP problems which have equality constraints. In this paper an asynchronous parallel evolutionary algorithm with scalable granularities for MIMD machines is designed by parallelizing the Robust Evolutionary Algorithm. A challenge NLP problem is chosen as the test function. Numerical experiments show that the new algorithm is very efficient and effective.

Keywords: parallel algorithm, Evolutionary Computation, function optimization

1. INTRODUCTION

There are efficient and effective algorithms for solving function optimiza- tion, such as genetic algorithms and evolution strategies [2]-[3]. Recently Tao Guo[1] proposed a new population random search algorithm which is based on the sub-space search (a general multi-parent recombination search strategy) combining with population hill-climbing for solving function optimization problems with inequality constraints. Li and others [4] propose a robust evolutionary algorithm, which is an extension of Guo's algorithm [1], which possesses enhanced capabilities for solving NLP problems. These capabilities include: a) advancing the variable subspace, b) adding a search process over subspaces and normalized constraints, c) using an adaptive penalty function, and d) adding the ability to deal with integer NLP problems, 0-1 NLP problems, and mixed-integer NLP problems which have equality constraints. These four enhancements increase the capabilities of the algorithm to solve nonlinear programming problems in a more robust and universal way. By parallelizing this algorithm, we get an asynchronous parallel evolutionary algorithm with scalable granularities for MIMD machines. Numerical experiments show that new algorithm is very efficient and effective for solving hard nonlinear function optimization problems by parallel and distributed computers.

The rest of the paper is organized as follows. In section 2, Li's algorithm [4] is introduced. An asynchronous parallel evolutionary algorithm with scalable granularities for MIMD machines is proposed in section 3. Finally in section 4, some numerical results are given.

2. LI'S ALGORITHM

The general Nonlinear Programming (NLP) problem can be expressed in the following form:

$$\text{Minimize } f(X,Y) \tag{1}$$

Subject to:

$$h_i(X,Y)=0 \qquad i=1,2,\ldots,k_1$$
$$g_j(X,Y)\le 0 \qquad j=1,2,\ldots,k_2$$
$$X^{lower} \le X \le X^{upper}$$
$$Y^{lower} \le Y \le Y^{upper}$$

where $X\in R^p$, $Y\in Z^q$, and the objective function $f(X,Y)$, the equality constraints $h_i(X,Y)$ and the inequality constraints $g_j(X,Y)$ are usually nonlinear functions which include both real variables X and integer variables Y.

Denoting the domain $D=\{(X,Y)\mid X^{lower} \le X \le X^{upper},\ Y^{lower} \le Y \le Y^{upper}, X\in R^p, Y\in Z^q\}$, and $D^*=\{(X, Y^*)\mid X^{lower} \le X \le X^{upper}, Y^{lower} \le Y^* \le Y^{upper}, X\in R^p, Y^*\in R^q\}$, we introduce the concept of a subspace V of the domain D^*. m points $Z_j=(X_j,Y_j^*)$, $j=1,2,\cdots,m$ in D^* are used to construct the subspace V, V, defined as

$$V=\{Z\mid Z=\sum_{i=1}^{m} a_i Z_i\}$$

where a_i is subject to $\sum_{i=1}^{m} a_i = 1, -0.5 \le a_i \le 1.5$ and $Z=(X, Y^*)$, where $Z\in D^*$

We define the fitness function as:

$$F(Z)=f(X,\text{int}(Y^*)) + r(t)\sum_{i=1}^{k_1}(\bar{h}_i(X,\text{int}(Y^*)))^2$$

where int (Y^*) is a integer function of Y^* which takes integer

part of Y^*, $\bar{h}_i(Z)$ denotes the normalized function of $h_i(Z)$ and $r(t)$ is a function of iteration number t.

Denoting

$$w_i(Z)=\begin{cases}0, & g_i(X,\text{int}(Y^*))\le 0\\ g_i(X,\text{int}(Y^*)), & \text{otherwise}\end{cases}$$

and $W(Z)=\sum_{i=1}^{k_3} w_i(Z)$,

we define the Boolean function "*better*" as follows:

better(Z_1,Z_2) =

$$\begin{cases}W(Z_1)\le W(Z_2) & \text{TRUE}\\ W(Z_1)>W(Z_2) & \text{FALSE}\\ (W(Z_1)=W(Z_2))\wedge(F(Z_1)\le F(Z_2)) & \text{TRUE}\\ (W(Z_1)=W(Z_2))\wedge(F(Z_1)>F(Z_2)) & \text{FALSE}\end{cases}$$

If better (Z_1,Z_2) is TRUE, this means that the individual Z_1 is better than the individual Z_2.

The Li's algorithm can now be described as follows.

Li's Algorithm
Begin
initialize $P = \{Z_1,Z_2,\ldots,Z_N\}$;
$Z_i \in D^*$;
$t := 0$;
$Z_{best} = \arg \min_{1\le i\le N} F(Z_i)$;
$Z_{worst} = \arg \max_{1\le i\le N} F(Z_i)$;
while not abs ($F(Z_{best}) - F(Z_{worst})$) $\le \varepsilon$ **do**
select randomly m points $z_1', z_2', \cdots, Z_m'$ from P to form the subspace V;
select s points *randomly* $Z_1^*, Z_2^*, \cdots, Z_s^*$ from V;
$Z' = \arg \min_{1\le i\le s} F(Z_i^*)$;
if *better* (Z', Z_{worst}) **then** $Z_{worst} := Z'$;
$t := t + 1$;
$Z_{best} = \arg \min_{1\le i\le N} F(Z_i)$;
$Z_{worst} = \arg \max_{1\le i\le N} F(Z_i)$;
if abs $(F(Z_{best}) - F(Z_{worst})) \le \eta$ **.and.** $m \ge 3$ **then** $m := m - 1$;
endwhile
output t, P ;
End

Where $Z_{best} = \arg \min_{1\le i\le N} F(Z_i)$ means that Z_{best} denotes the argument Z_j which satisfies $\min_{1\le i\le N} F(Z_i)$.

The algorithm has the two important features:

1. The ergodicity of the search. During the random search of the subspace, we employ a "non-convex combination" approach, that is, the coefficients a_i of $Z' = \sum_{i=1}^{m} a_i Z_i$ are random numbers in the interval [-0.5, 1.5]. This ensures a non-zero probability that any point in the solution space is searched. This ergodicity of the algorithm ensures that the optimum is not ignored.

2. The monotonic fitness decrease of population (when the minimum is required). Each iteration $(t \to t+1)$ of the algorithm discards only the individual having Worst fitness in the population. This ensures a monotonically decreasing trend of the values of objective function of the population, which ensures that each individual of the population will reach the optimum.

When we consider the population P (0), P (1), P (2),..., P (t),... as a Markov chain, following the way of [5] one can prove the convergence of the algorithm.

3. ASYNCHRONOUS PARALLEL EVOLUTIONARY ALGORITHMS

Li's algorithm is a sequential algorithm, but we can parallelize it as an asynchronous parallel evolutionary algorithm.

We define a parallel algorithm for multiprocessors (or multicomputers) as a collection of concurrent processes that may operate simultaneously for solving a given problem [5].

An asynchronous parallel algorithm is a parallel algorithm of the following properties [6]:

1. There is a set of global variables accessible to all processes.

2. When a stage of a process is complete, the process first reads some global variables. Then based on the values of the variables together with the results just obtained from the last stage, the process modifies some global variables, and then activates the next stage or terminates itself. In many cases, to ensure logic correctness, the operations on global variables are programmed as critical sections.

The global variables (shared data) are stored in the shared memory of the MIMD machine or in some local memory of the multiprocessor system.

Denote the population of solutions of problem (1) by $P = \{Z_1,Z_2,\cdots,Z_N\}$, where individual $Z_j = (X_j, Y_j^*)$, $X_j \in R_p$ and $Y_j^* \in R_q$, and $(X_j, \text{int}(Y_j^*))$ is a candidate solution of problem (1) which is represented by vector (X, Y). We assume that a population of N individuals is assigned to each of processors of a multiprocessor system. Each processor executes the same program PROCEDURE APEA (Asynchronous Parallel Evolutionary Algorithm) to steer the asynchronous parallel computation.

PROCEDURE APEA
Begin
initialize $P = \{Z_1,Z_2,\ldots,Z_N\}$; $Z_i \in D^*$;
$t := 0$;

$Z_{best} = \arg \min_{1 \le i \le N} F(Z_i)$;

$Z_{worst} = \arg \max_{1 \le i \le N} F(Z_i)$;

while not abs ($F(Z_{best}) - F(Z_{worst})$) $\leqslant \varepsilon$ **do**

select randomly m points $Z_1', Z_2', \cdots,$

Z_m' from P to form the subspace V;

select s points *randomly* $Z_1^*, Z_2^*, \cdots,$

Z_s^* from V;

$Z' = \arg \min_{1 \le i \le r} F(Z_i)$;

if *better* (Z', Z_{worst}) **then** $Z_{worst} := Z'$;

$t := t + 1$;

$Z_{best} = \arg \min_{1 \le i \le N} F(Z_i)$;

$Z_{worst} = \arg \max_{1 \le i \le N} F(Z_i)$;

if ($t \equiv 0$ (mod T)) **then** broadcast

Z_{best} to Q neighbors;

while (any received message probed) **do**

if better (recv−individual , Z_{best}) **then** Z_{best}:= recv-individual

else Z_{worst} := recv-individual;

$Z_{worst} = \arg \max_{1 \le i \le N} F(Z_i)$;

endwhile}

if abs $(F(Z_{best}) - F(Z_{worst})) \leqslant \eta$ **.and.** $m \geqslant 3$ **then** $m := m - 1$;

endwhile

output t , P ;

End

where $t \equiv 0$ (mod T) denotes that t is congruent to zero with respect to modulus T which determines the frequency of migration between the processors.

Remark 1. The asynchronous communication between processors is implemented by calling **pvm-mcast ()**, **pvm-prob** and **pvm-nrecv ( )** which are provided by PVM.

Remark 2. T determines the computational granularity of the algorithm, and together with Q, the number of neighbor processors to communicate with, control the cost of communication. That is why the granularity of the algorithm is scalable. For more details of the algorithm, interested readers can refer to [4].

4. NUMERICAL EXPERIMENTS

For testing the parallel efficiency and effectiveness of our new algorithm, a challenge NLP problem called Bump Problem introduced by Keanne [6] is chosen as the test problem:

$$\text{Maximize } f_n(x) = \frac{\left| \sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i) \right|}{\sqrt{\sum_{i=1}^{n} i x_i^2}}$$

$$\text{Subject to } 0 < x_i < 10 \ , \ 1 \le i \le n \ , \quad \prod_{i=1}^{n} x_i \ge 0.75 \ ,$$

$$\sum_{i=1}^{n} x_i \le 7.5n$$

Because the BUMP problem is super-nonlinear, super-multimodal (multi-peaked), and super-multidimensional (if n is large enough), it has been used as a universal test function to compare the performances of optimization algorithms from around the world. Fig.1 illustrates its objective function surface when $n = 2$. Since the constraint $\prod_{i=1}^{n} x_i \ge 0.75$ is essential, we replace it with the equality constraint $\prod_{i=1}^{n} x_i \ge 0.75$ to test the performance of our algorithm.

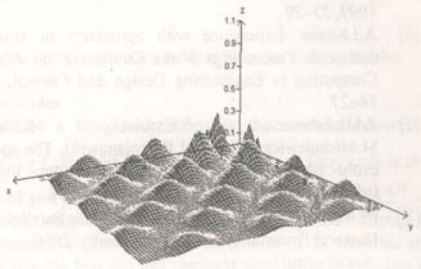


Fig.1 The f2 landscape

For making a comparison with other algorithms we take a 50-dimensional bump problem, which is, the largest dimensional problem solved so far as the example. In this case, setting the parameters at N =50, m =7, r =100000, s = 8, η =10^{-3}, ε =10^{-14} and K = 1, we obtain the following result (with an accuracy of 1 part in 10^{14}) : $f_{50}(x^*) = 0.83526201238794$. The optimal value of this problem is still unknown, but our result is the best to be published so far. See [7].

When n = 10000, setting the parameters at N = 10000, m = 10, r =100000, s = 8, $\eta = 10^{-3}$, $\varepsilon = 10^{-14}$, T = 300, Q = 10 and K = 256 we obtain the best result: $f_{10000}(X^*) = 0.84564074936999$. The numerical experiments are performed on a massively parallel computer system (an MIMD machine).

Because the algorithm is performed in an asynchronous parallel way, the processors need not to wait for each other, and added another stochastic factor (some individuals migrate randomly between processors), so the phenomena of super-linear speed-up have been observed. For more details about the experiments of the Bump Problem, interested readers can refer to Liu's doctoral dissertation [8].

Acknowledgement

This work was supported in part by National Science Foundation of China (No. 70071042 and No.60073043) and National Laboratory for Parallel and Distributed Processing.

REFERENCES

[1] Tao Guo, Lishan Kang. A new evoluTionary algorithm for function optimal Zation Wuhan University Journal of

Nature Science. Vol. 4 No.4 1999

[2] 409~414Kalyanmoy Deb. GeneAS: A robust optimal design technique for mechanical component design. Evolutionary algorithm in engineering application: Springer-Verlag,1997, 497~514

[3] Carlos A. Coello: Self-adaptive penalties for GA-based optimization, in Proceedings of the Congress on Evolutionary Computation, Washington, D.C. USA, IEEE Press, 1999,537~580

[4] Yan LI, Lishan KANG, Hugo de GARIS, Zhuo KANG, Pu LIU, A Robust Algorithm for Solving Nonlinear Programming Problems (submitted to Intern. J. of Computer Math.)

[5] Jun He, Lishan Kang. On the convergence rates of genetic algorithms. Theoretical Computer Science, 229, 1999, 23~29

[6] A.J.Keane: Experience with optimizers in structural design, in Proceedings of the Conference on Adaptive Computing in Engineering Design and Control, 1994, 14~27

[7] Z.Michalewicz, S.Esquivel, R.Gallard, M.Michalewicz,T.Guo and K.Trojanowski, The spirit of evolu- tionary algorithms, Journal of Compu- ting and Information Technology-CIT 7, 1999,1, 1-18.

[8] P.Liu, Evolutionary Algorithms and Their Parallelization, Doctoral Dissertation, Wuhan University, 2000.

On the Evolutionary Technique of Bisection and Its Applications*

Nengchao Wang
Parallel Computation Institute, Huazhong University of Science and Technology
Wuhan, Hubei 430074, China
Email: ncwang@public.wh.hb.cn
and
Qing Li
College of Computer Science and Technology, Huazhong University of Science and Technology
Wuhan, Hubei 430074, China
Email: qingli@public.wuhan.cngb.com

ABSTRACT

A method called bisection evolutionary method for the designing of algorithm is proposed in this paper. With this method, some fast transform algorithms and parallel algorithms have been discussed.

Keywords: Bisection method, Doubling method, Gray code, Hypercube, Fast Fourier transform

1. INTRODUCTION

It is well known that the *doubling method* [1] is a basic technique for designing synchronous parallel algorithms. This method has met significant success for parallel computations since 70's last century. However, in general, the design of a doubling algorithm for certain problem seems to require "some sort of magic" because some auxiliary functions must be introdueed. For example, consider the following one-order iteration

$$\begin{cases} x_0 = b_0 \\ x_i = a_i x_{i-1} + b_i \quad i = 1,2,\cdots,N-1 \end{cases} \tag{1.1}$$

The two auxiliary functions are needed:

$$Q(j,i) = \sum_{l=i}^{j} \left(\prod_{k=l+1}^{j} a_k \right) b_l, \qquad P(j,i) = \prod_{k=i}^{j} a_k .$$

One can derive some important properties of the functions, and then gets

$$\begin{cases} x_{2i} = \left(\prod_{k=i+1}^{2i} a_k \right) Q(i,1) + Q(2i, i+1) \\ x_{2i+1} = \left(\prod_{k=i+1}^{2i+1} a_k \right) Q(i,1) + Q(2i+1, i+1) \end{cases}$$

One can also see that the procedure of algorithm design is tedious.

In this paper, we recommend a method called bisection evolutionary method for the designing of algorithm including fast transform algorithms, parallel algorithms, and network architectures.

2. BISECTION EVOLUTIONARY METHOD

In this section, we give the basic idea and basic operations of the bisection evolutionary method by using some examples.

Gray Codes

To define a Gray code, consider the numbers between 0 and $2^n - 1$ by their binary representation. A Gray code is then defined as a permutation of the numbers between 0 and $2^n - 1$ such that neighboring elements have exactly one differing bit (so successive elements differ by powers of 2). Here we also mean that the first and last elements must differ in only one bit position. For example, when $n = 2$ the $\{0,1,3,2\}$ is a Gray code. There are many ways of generating Gray codes and they have been studied for several applications.

We consider the binary reflected Gray code. It is generated by the following procedures for $k = 1,2,3,\cdots$.

- Initial state: $G_1 = \{g_0, g_1\} = \{0,1\}$
- Dichotomy procedure

D_0 : Copy G_k and put 0's at the front, that is,

$$G_k^{(0)} = \{0g_0, 0g_1, \cdots, 0g_{2^k-1}\}$$

D_1 : Copy G_k and reversed it then put 1's at the front, that is, $G_k^{(1)} = \{1g_{2^k-1}, 1g_{2^k-2}, \cdots, 1g_0\}$

- Combination procedure

$$C_{0-1}: \quad G_{k+1} = \{G_k^{(0)}, \quad G_k^{(1)}\}$$

That is

$G_1 = \{ \quad 0, \quad 1\}$
$G_2 = \{ \ 00, \ 01, \ 11, \ 10\}$
$G_2 = \{000, 001, 011, 010, 110, 111, 101, 100\}$
$\vdots$

or

$G_1 = \{0, 1\}$
$G_2 = \{0, 1, 3, 2\}$
$G_2 = \{0, 1, 3, 2, 6, 7, 5, 4\}$
$\vdots$

Hypercube

To describe a hypercube architecture assumes that the processors are labeled from 0 to $2^n - 1$ for some value of n. In a hypercube two processors are linked if and only if their binary representation differ in exactly one bit positions. Thus the

* This work was supported by National Science Foundation, Grant No. 60073044.

indices of neighboring processors differ by a power of 2 (but the converse does not hold). Another way to regard a hypercube is iteratively: a 0-dimensional hypercube is just one processor and for k greater than 1 we define a k-dimensional hypercube as two (k-1)-dimensional hypercubes with links between corresponding processors in each half. And this procedure is a bisection evolution.

- Initial state: H_0 (one processor)
- Dichotomy procedure

  D_0 : Copy H_k and put 0's at the front each node's number

  D_1 : Copy H_k and put 1's at the front each node's number
- Combination procedure

  C_{k-1} : Link the homologous nodes to form H_{k+1}

We show the procedure through diagrams below

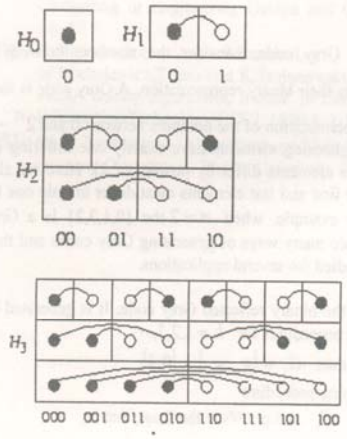


Fig. 1. Hypercubes

Now, we use the following figure to indicate the bisection evolutionary systems.

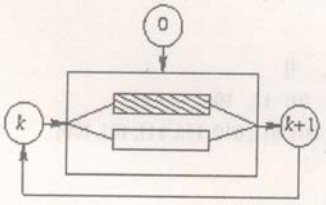


Fig. 2. Bisection evolutionary system

In Fig. 2, "0" stands for initial system state. The changes of the system state between two time steps involve two procedures ------ dichotomy and combination [2].

The One-Order Iteration

The Eqs. (1.1) can be regarded as a chain with $N=2^n$ nodes (the step-lengths are all 1)

$$L_1: \ x_0 \to x_1 \to \cdots \to x_{N-1}$$

With the bisection evolutionary procedure, we can separate the L_1 into two chains: an even chain and an odd chain and then combine them to form a new vector chain L_2 with nodes $N/2$ (the step-lengths are all 2).

$$L_2: \ \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \to \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} \to \cdots \to \begin{bmatrix} x_{N-2} \\ x_{N-1} \end{bmatrix}$$

Repeat this procedure and L_n is the solution. So we have

Algorithm 1. For $k=1,2,\cdots,n$ compute

$$a_i^{(k)} = a_i^{(k-1)} a_{i-2^{k-1}}^{(k-1)} \quad i = 2^k, 2^k+1, \cdots, N-1$$

$$b_i^{(k)} = \begin{cases} b_i^{(k-1)}, & i = 0,1,\cdots,2^{k-1}-1 \\ b_i^{(k-1)} + a_i^{(k-1)} b_{i-2^{k-1}}^{(k-1)}, & i = 2^{k-1},\cdots,N-1 \end{cases}$$ Then

$x_i = b_i^{(n)}, \ i = 0,1,\cdots N-1$.

3. FAST FOURIER TRANSFORM

The Discrete Fourier Transform (DFT) of a sequence $\mathbf{x} = (x_0, x_1, \cdots, x_{N-1})$ of size N is the sequence $\mathbf{X} = (X_0, X_1, \cdots, X_{N-1})$, of size N, defined by:

$$X_j = \sum_{k=0}^{N-1} x_k W_N^{jk}, \quad j = 0,1,\cdots,N-1$$

Where $W_N = \exp(-2i\pi/N)$, and $i = \sqrt{-1}$.

The first algorithm of FFT was proposed by Cooley and Tukey. The FFT algorithm uses a greatly reduced number of arithmetic operations as compared to the brute force computation of the DFT. Since then, a lot of variation has been deducted from the original algorithm. They only differ in the way of storing intermediate data. The basic idea of these algorithms is the splitting of data entry $\mathbf{x}$ into two subsets at each step of the algorithm, and combines them using a butterfly scheme [3].

Look-Up Tables of Twiddle Factors

For implementation of FFT algorithms, one can create an array to store the twiddle factors, and this array is called look-up table. Let the number of points in the data sequence be a power of 2, that is, $N=2^n$, where n is an integer. It is clear that $W_N^0 = W_N^N = 1, W_N^{N/2} = -1$, and $W_{kN}^{kj} = W_N^j$. So the size of the array only needs $N/2$ complex units.

Besides of the ***natural order*** look-up table

$$\mathbf{U}_N = (W_N^0, W_N^1, \cdots, W_N^{N/2-1}),$$

We recommend the following ***bit-reversed order*** (BRO) look-up table

$$\mathbf{V}_N = (W_N^{<0>_{n-1}}, W_N^{<1>_{n-1}}, \cdots, W_N^{<N/2-1>_{n-1}}),$$

where $< j >_{n-1} = (j_0 j_1 \cdots j_{n-3} j_{n-2})$ if $j = (j_{n-2} j_{n-3} \cdots j_1 j_0)$.

Theorem 1. For $N = 2,4,8,16,\cdots$, we have the following recursion formula

$$\begin{cases} \mathbf{V}_2 = (1) \\ \mathbf{V}_{2N} = (\mathbf{V}_N, \ W_{2N} \times \mathbf{V}_N) = (1, \ W_{2N}) \otimes \mathbf{V}_N \end{cases}$$

Proof:

For $0 \le j < N/2$, $j = (0 j_{n-2} j_{n-3} \cdots j_1 j_0)$, we have

$< j >_n = (j_0 j_1 \cdots j_{n-3} j_{n-2} 0) = 2 < j >_{n-1}$,
and
$< j+N/2 >_n = (j_0 j_1 \cdots j_{n-3} j_{n-2} 1) = 2 < j >_{n-1} + 1$.
Consider the components of $\mathbf{V}_{2N}$, we can get

$$v_{2N}(j) = W_{2N}^{<j>_n} = W_{2N}^{2<j>_{n-1}} = W_N^{<j>_{n-1}} = v_N(j)$$

and

$$\begin{aligned} v_{2N}(j+N/2) &= W_{2N}^{<j+N/2>_n} = W_{2N}^{2<j>_{n-1}+1} \\ &= W_{2N} W_N^{<j>_{n-1}} = W_{2N} v_N(j). \end{aligned}$$

The BRO look-up table of twiddle factors can be generated evolutionarily by bisection procedure.

- Initial state: $\mathbf{V}_2 = (1)$
- Dichotomy procedure
  D_0 : Copy $\mathbf{V}_k$
  D_1 : Copy $\mathbf{V}_k$ and multiplied by W_{2k}
- Combination procedure
  C_{0-1} : $\mathbf{V}_{2k} = (\mathbf{V}_k, W_{2k} \times \mathbf{V}_k)$.

The theorem 1 indicates that the table is extendible. That is, we can compute all 2^m-FFT $(m \le n)$ if $\mathbf{V}_N$ is ready. We can get the BRO table using a clever scheme (see [4]). Note, the first 4 components of $\mathbf{V}_N$ are 1, $-i$, $(1-i)/\sqrt{2}$, and $-(1+i)/\sqrt{2}$. It is trivial for a complex number multiplies $v(i)$, $(i = 0, 1, 2, 3)$.

FFT Algorithms Based on BRO Twiddle Factors Look-UP Table

With binary representation, we have the typical FFT algorithm.

Algorithm 2. Let $x_0(k_0 k_1 \cdots k_{n-2} k_{n-1}) = x(k_{n-1} k_{n-2} \cdots k_1 k_0)$. For $l = 1, 2, \cdots, n$, do

$$x_l(k_0 \cdots k_{n-l-1} j_{l-1} \cdots j_0) = \sum_{k_{n-l}=0}^{1} x_{l-1}(k_0 \cdots k_{n-l} j_{l-2} \cdots j_0) W_N^{j_{l-1} 2^{l-1} \sum_{r=0}^{n-l} k_r 2^r}$$

Then $X(j_{n-1} j_{n-2} \cdots j_1 j_0) = x_n(j_{n-1} j_{n-2} \cdots j_1 j_0)$.

The flow diagram of algorithm 2 for $N = 8$ is shown in Fig. 3.

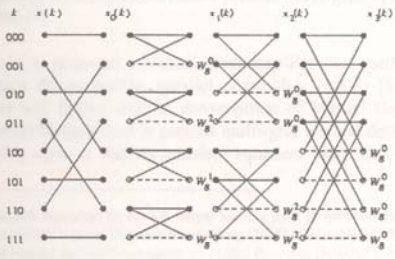


Fig. 3. Flow diagram for algorithm 2

It is easy to count that the computational complexity of algorithm 2 is $5N \log_2 N - 10N + 16$, and about $N \log_2 N$ extra auxiliary operations are needed for computer computation.

Merging all two steps of algorithm 2, we get the following radix-4 based FFT algorithm.

Algorithm 3. Let $x_0(k_0 k_1 \cdots k_{n-2} k_{n-1}) = x(k_{n-1} k_{n-2} \cdots k_1 k_0)$, For $l = 1, 3, 5, \cdots$ until $l \le n-1$, do

$$x_{l+1}(k_0 \cdots k_{n-l-2} j_l \cdots j_0) = \sum_{k_{n-l-1}=0}^{1} \sum_{k_{n-l}=0}^{1} x_{l-1}(k_0 \cdots k_{n-l} j_{l-2} \cdots j_0) P_l,$$

here $P_l =$

$$(-1)^{j_{l-1} k_{n-l} + j_l k_{n-l-1}} (-i)^{j_{l-1} k_{n-l-1}} [v(2K_l)]^{j_{l-1}} [v(K_l)]^{j_l}$$ and

$K_l = (k_0 k_1 \cdots k_{n-l-2})$.

If n is odd, do

$$x_n(j_{n-1} j_{n-2} \cdots j_1 j_0) = \sum_{k_0=0}^{1} x_{n-1}(k_0 j_{n-2} \cdots j_1 j_0)(-1)^{k_0 j_{n-1}}$$

then $X(j_{n-1} j_{n-2} \cdots j_1 j_0) = x_n(j_{n-1} j_{n-2} \cdots j_1 j_0)$.

Numerical results

Table 1 shows the number of the arithmetic operations including additions and multiplications, a_N, and all of the auxiliary operations, b_N of algorithm 3. We can see that

$c_N = a_N / (N \log_2 N) < 4$,
and
$d_N = (a_N + b_N) / (N \log_2 N) < 5$.

Tab. 1. Number of operations of Algo.3

N	*a*	*b*	*c*	*d*
8	56	55	2.3	4.6
16	168	101	2.6	4.2
32	464	265	2.9	4.6
64	1184	507	3.1	4.4
128	2912	1279	3.3	4.7
256	6880	2449	3.4	4.6
512	15968	5941	3.5	4.8
1024	36192	11399	3.5	4.7
2048	81248	26923	3.6	4.8
4096	179552	51837	3.7	4.7
8192	394592	120097	3.7	4.8
16384	857440	232051	3.7	4.8
32768	1856854	529687	3.8	4.8
65536	3986784	1026665	3.8	4.8
131072	8541536	2315533	3.8	4.8
262144	18175328	4500063	3.9	4.8
524288	38622560	10048771	3.9	4.9
1048576	81614176	19573333	3.9	4.8

Our numerical experiments are performed on a PC (Pentium, CPU at 400MHz, RAM 128MB), under Linux operation system (Red Hat Linux V6.1). The code is written with C language. For $N = 2^n$, we compare the elapsed times between our algorithm 3 and the FFTW (Fastest Fourier Transforms in the

West)[5] software package.

In the following table 2, t_1 and t_2 represent the elapsed time (in seconds) of algorithm 3 and FFTW software package respectively. The times in table 2 include the time of computation of forward and backward Fourier transforms.

Tab.2. Comparison between algorithm 3 and FFTW

N	t_1	t_2
8	0.000004	0.000053
16	0.000007	0.000082
32	0.000016	0.000126
64	0.000029	0.000213
128	0.000064	0.000346
256	0.000132	0.000600
512	0.000319	0.001073
1024	0.000679	0.002050
2048	0.001679	0.004269
4096	0.003835	0.011648
8192	0.016406	0.028965
16384	0.070460	0.071353
32768	0.161034	0.160896
65536	0.338422	0.331420
131072	0.711670	0.704253
262144	1.485745	1.606859
524288	3.107343	3.345176
1048576	6.474286	7.017258

4. CONCLUSION

In this paper we have introduced the bisection evolutionary technique for designing some kinds of algorithms. This method is easy to handle, easy to understand. The structures of the algorithms designed by this technique are more symmetry and those algorithms are efficient.

REFERENCES

[1] P. M. Kogge and H. S. Stone, "A parallel algorithm for the efficient solution of a general class of recurrence equations", IEEE Trans. Comp., Vol.22, 1973, pp.786-792.

[2] N. Wang, "Bisection technique for designing synchronous parallel algorithms". Science in China. Ser. A, Vol.38, No.5, 1995, pp.635-540.

[3] A. M. Grigoryan and S. S. Agaian, "Split manageable efficient algorithm for Fourier and Hadamard transforms", IEEE Trans. Signal Processing, Vol. 48, 2000, pp.172-183

[4] J. M. Rius and R. D. Porrata-Doria, "New FFT bit-reversal algorithm", IEEE Trans. Signal Processing, Vol. 43, 1995, pp.991-994.

[5] http://www.fftw.org

Theory of Virtual Boundary Forecast Multi-Grid Method for Parallel Calculation of Transient Equation*

Guo Qingping[1] Wei Jialin[1] Yakup Paker[2]
Dennis Parkinson[2] Zhang Shesheng[1]
[1] Wuhan Transportation University, Wuhan 430063, P. R. China
[2]Queen Mary & West Field College, University of London E1 4NS U.K.
qpguo@public.wh.hb.cn paker@dcs.qmw.ac.uk

ABSTRACT

This paper proposes and theoretically proves a Virtual Boundary Forecast (VBF) multi-grid method for parallel calculation of transient equation. Some forecast theorems, such as necessary and sufficient conditions of VBF convergence, have been proven with some examples. Using our new method, a quite impressive speed up and higher computer parallel efficiency for solving transient equation have been obtained.

Keywords: Domain Decomposition, Virtual Boundary Forecast, Multi-grid Method, Network-based Parallel Computing

1. INTRODUCTION

In general, numerical solution of non-linear equations requires more times of iteration than linear ones. There are many kinds of iterative methods. For example, Wang et al (1996) [1] solved a non-linear Poisson equation by using a simple iterative method, requiring 18023 iterations to obtain converged results, which took too long time. People need fast convergence iterative methods for solving non-linear equations. The multi-grid algorithm has faster convergence speed than simple iterative method used before. An efficient parallel multi-grid algorithm used on a local network could obtain a solution for non-linear equations with less time. Guo et al proposed an optimum parallel multi-grid algorithm used on a local network (2000) [2]. Generally speaking, a domain of initial-boundary problem in parallel calculation must be decomposed into several sub domains. Hence some virtual boundaries are created along sub domains. Data are transferred between sub domains by means of communication between computers. The total computation time in some sense depends on communication frequency (1990) [3]. Matheson pointed out that there should have some method to reduce communication frequency for designing efficient parallel multi-grid algorithms (1996) [4].

Guo et al proposed a Virtual Boundary Forecast method for domain decomposition parallel computing (2000) [5]. This paper will further discuss convergence of Virtual Boundary Forecast Method used in parallel multi-grid method design for solving static as well as transient equation. Key idea of this approach is to reduce communication frequency between sub domains by increasing a small amount of computation in virtual boundary forecast on each sub domain; such that obtain higher parallel speedup.

2. VBF METHOD

Before proving convergence of Virtual Boundary Forecast Method in solving static and transient equations with parallel multi-grid method, we first describe the idea and algorithm steps of VBF method. Let us take a one-dimension problem as example.

Suppose one dimension non-linear initial-boundary problem given as

$$\begin{aligned} &L^{(\Omega)}u = f^{(\Omega)}(t, x, u) \quad u = u(x, t) \quad x \in \Omega \\ &L^{(\Gamma)}u = f^{(\Gamma)}(t, x, u) \quad x \in \Gamma \\ &u(x, 0) = g(x) \quad x \in \Gamma \end{aligned} \tag{1}$$

Here Γ is boundary of domain Ω. We decompose Ω into P sub-domains Ω_k $k = 1,2...P$; Γ_k is virtual boundary of Ω_k, that excludes the actual boundary Γ. The whole virtual boundary can be written as:

$$\Gamma_v = \Gamma_1 + \Gamma_2 + ... + \Gamma_P$$

On the grid $\Omega^{(h)}$, Eq.(1) can be written as a difference formula

$$U = F(U, t)$$

and iterative formula

$$U^{(s+1)} = F(U^{(s)}, t) \tag{2}$$

where $U = (u_1, u_2, ...u_n)^T$

$U^{(s)} = (u_1^{(s)}, u_2^{(s)}, ..., u_n^{(s)})^T$

Suppose $u_{hk} = u_{hk}(x)$ are numerical value of function u on the virtual boundary for k^{th} iteration and $x \in \Gamma_v$, then $\{u_{hk}\}$ is a set of these values. We construct a subspace R^m defined as

$$R^m = \{u_{h1}, u_{h2}, ..., u_{hm}\}$$

Assume G is a mapping of R^m to R

$$V^{(m+1)} = G(Y) \quad Y \in R^m$$

$V^{(m+1)}$ can be taken as virtual boundary forecast value of function u on $(m+1)^{th}$ iteration. The forecast procedures are:

(1). Decomposition of domain Ω into sub-domains Ω_k $k = 1,2,...,P$; and

$$V_k = U(X_k) \qquad X_k = \Omega^{(h)} \in \Omega_k$$

(2). Given initial iterative value $V_k^{(0)}$, relative virtual boundary values b_0, error bounds ε, finest grid $\Omega^{(h)}$ and coarsest grid $\Omega^{(H)}$, integer N_{min}, initial iterative number $s = 1$;

(3). From Eq.(2) using normal parallel multi grid method on

* Research supported by the UK Royal Society joint project (the Royal SocietyQ724), the Natural Science Foundation of China (NSFC Grant No.69773021) and the Natural Science Foundation of Hubei Province (NSFHP 2000 J153).

the coarsest grid calculate $V_k^{(1)}$, a relative virtual boundary value b_1 is obtained; $s = s+1$, doubling the grid density then $V_k^{(2)}$ and value b_2 obtained;

(4). $s = s+1$; forecast b_3 with some forecasting methods. In this example the quadratic polynomial limited value formula is used [2], that is

$$b_3 = b_2 - \frac{a^2}{4b}$$

Here $a = 0.5\,(b_2 - b_0)$, $b = 0.5\,(b_2 - 2b_1 + b_0)$

(5). Take b_3 as the virtual boundary value, calculate function value $V_k^{(3)}$ on sub domain Ω_k, $k = 1, 2, \ldots P$; in this step there is no need of communication between sub domains, multi grid calculation only takes place inside sub domain; the detailed multi grid calculation will be shown in the next section.

(6). Using Eq.(2), calculate $V_k^{(4)}$ on the whole domain Ω, in this step we do not use the multi grid method, but only use smooth iteration method. That means communication must take place between sub domains and relative virtual boundary values denoted as b_4 are updated;

(7). If $s < N_{min}$ then rewrite $b_4 \to b_2$, $b_3 \to b_1$, $b_2 \to b_0$, go to step (4); otherwise do next step;

(8). If $|V_k^{(4)} - V_k^{(3)}| < \varepsilon$, go to step (9), otherwise rewrite $b_4 \to b_2$, $b_3 \to b_1$, $b_2 \to b_0$, go to step (4);

(9). End iteration calculation on sub domain Ω_k, output numerical results of u, and end communication between Ω_k and its neighbouring sub domain;

(10). If function u converges over all sub domains then we end calculation on the whole domain Ω.

3. THEOREMS AND PROPFS

General proof of convergence of the VBF method in parallel calculation is very difficult. In this section some forecast theorems, such as necessary and sufficient conditions of VBF convergence, have been proven with some examples.

Suppose $U = (u_1, u_2, \ldots u_n)^T$ is a vector of n dimensional space R^n, $A = (a_{ij})_{n\times n}$ is n×n matrix, $\rho(A)$ is maximal absolute value of characteristic root of matrix A.

Suppose $U^{(s)} = (u_1^{(s)}, u_2^{(s)}, \ldots u_n^{(s)})^T$ satisfies the following iterative formula

$$U^{(s+1)} = A\,U^{(s)} \quad s=0,1,2,\ldots$$

Example 1. Static equation

$$U'' = 0 \quad 0<x<1$$
$$u(0) = 0 \quad u(1) = 1 \tag{3}$$

A finite difference formula of above equation is easily derived as:

$$u_0 = 0$$
$$u_n = 0.5(u_{n-1} + u_{n+1}) \quad n = 1,2,\ldots,N$$
$$u_{N+1} = 1$$

The iterative formulae are:

$$u_0^{(s)} = 0$$
$$u_n^{(s+1)} = 0.5(u_{n-1}^{(s)} + u_{n+1}^{(s)}) \quad n = 1,2,\ldots,N \tag{4}$$
$$u_{N+1}^{(s)} = 1 \quad s = 0,1,2,\ldots$$

and the initial values of iterative are taken as

$$u_n^{(0)} = 0 \quad n = 0,1,\ldots,N \qquad u_{N+1}^{(0)} = 1 \tag{5}$$

Theorem 1. If $u_n^{(s)}$ satisfies Eq.(4) and Eq.(5), then it can be concluded that:

(a). $u_n^{(1)} - u_{n-1}^{(1)} < 0$ (b). $2u_n^{(1)} - u_{n-1}^{(1)} - u_{n+1}^{(1)} < 0$

Proof: From Eq.(4) and Eq.(5), It can be gotten that

$u_n^{(1)} = 0 \quad n = 0,1,2\ldots,N-1$

$u_N^{(1)} = 0.5, \quad u_{N+1}^{(1)} = 1$

so that $u_n^{(1)}$ satisfies the following formula:

$$u_n^{(1)} - u_n^{(0)} \geqslant 0$$
$$2u_n^{(1)} - u_{n-1}^{(1)} - u_{n+1}^{(1)} \leqslant 0 \qquad \#$$

Theorem 2. If $u_n^{(s)}$ satisfies Eq.(4) and Eq.(5), then it can be concluded that:

(a) $u_n^{(s)} - u_n^{(s-1)} \geqslant 0$ (b) $2u_n^{(s)} - u_{n-1}^{(s)} - u_{n+1}^{(s)} \leqslant 0$ (6)

Proof: From Theorem 1, Eq.(6) is correct when $s = 1$. Suppose Eq.(6) is correct when $n = k$, or

$$u_n^{(k)} - u_n^{(k-1)} \geqslant 0 \quad 2u_n^{(k)} - u_{n-1}^{(k)} - u_{n+1}^{(k)} \leqslant 0 \tag{7}$$

When $n = k+1$, from Eq.(4) and Eq.(7) we have

$$u_n^{(k+1)} - u_n^{(k)} = 0.5(u_{n-1}^{(k)} + u_{n+1}^{(k)}) - u_n^{(k)} = -0.5(2u_n^{(k)} - u_{n-1}^{(k)} - u_{n+1}^{(k)}) \geqslant 0$$

$$2u_n^{(k+1)} - u_{n-1}^{(k+1)} - u_{n+1}^{(k+1)} =$$
$$= 0.5[2(u_{n-1}^{(k)} + u_{n+1}^{(k)}) - (u_{n-2}^{(k)} + u_n^{(k)}) - (u_n^{(k)} + u_{n+2}^{(k)})]$$
$$= 0.5[(2u_{n-1}^{(k)} - u_{n-2}^{(k)} - u_n^{(k)}) + (2u_{n+1}^{(k)} - u_n^{(k)} - u_{n+2}^{(k)})] \leqslant 0 \ \#$$

Theorem 3. If $u_n^{(s)}$ satisfies Eq.(4) and Eq.(5), then it can be concluded that:

$$u_n^{(0)} \leqslant u_n^{(1)} \leqslant u_n^{(2)} \leqslant \ldots u_n^{(s)} \leqslant \ldots \leqslant u_n^{(*)} \qquad n = 1,2,\ldots N$$

Here $u_n^{(*)}$ is a solution of Eq.(4), or the limit value of the sequence $\{u_n^{(s)}\}$.

Proof: From theorem 1 and 2, it has been proven that

$$u_n^{(0)} \leqslant u_n^{(1)} \leqslant u_n^{(2)} \leqslant \ldots u_n^{(s)} \leqslant \ldots$$

Rewrite Eq.(4) as a matrix form:

$$U^{(s+1)} = AU^{(s)} + B \tag{8}$$

$U^{(s)} = (u_1^{(s)}, u_2^{(s)}, \ldots u_N^{(s)})^{(T)}$

$B = (0, 0, \ldots 0.5)$

$A = (a_{ij})$

Where

$a_{ii-1} = a_{i-1,i} = 0.5 \quad i = 2,3,\ldots N-1$

$a_{1,2} = a_{N,N-1} = 0.5$

$a_{ij} = 0$ for other i and j

As shown in [1] the maximal absolute value of the eigenvalue of matrix A is less than one or $\rho(A)<1$, so that Eq.(8) has a limit value

$U^{(*)} = (u_1^{(*)}, u_2^{(*)}, \ldots u_N^{(*)})^{(T)}$ or

$$Limt\ U^{(s)} = U^{(*)}$$
$$s \to \infty$$

For any s, we have

$u_n^{(s)} \leqslant u_n^{(*)} \quad n = 1,2,\ldots N \qquad \#$

Suppose the polynomial of degree 2 is

$$f_n(t) = u_n^{(s)} + a(t-s) + b(t-s)^{(2)} \tag{9}$$

and

$$f_n(s-1) = u_n^{(s-1)}, \quad f_n(s+1) = u_n^{(s+1)}$$

We can easily prove following theorem 4.

Theorem 4. $a = 0.5(u_n^{(s+1)} - u_n^{(s-1)})$

$b = 0.5(u_n^{(s-1)} - 2u_n^{(s)} + u_n^{(s+1)})$

Theorem 5. Given fixed n, for any $s>1$, and $u_n^{(s+1)} - u_n^{(s)} \geqslant 0$,

Eq.(9) has a maximal value

$$u_n^{(*)} = u_n^{(s)} - 0.25a^2/b \tag{10}$$

if and only if $u_n^{(s)}$ satisfy Eq.(6).

Proof: Sufficient condition.

Suppose $u_n^{(s)}$ satisfies Eq.(6), then $b<0$; as shown in [6], we can easily prove that Eq.(9) has a maximal value $u_n^{(*)}$ when $s>1$;

Necessary condition:

For any $s>0$, suppose Eq.(9) has a maximal value $u_n^{(*)}$, From paper [5] .
We have $u_n^{(s-1)} - 2u_n^{(s)} + u_n^{(s+1)} < 0$
So that $u_n^{(s)}$ satisfies Eq.(6). #

Example 2. Transient equation

$$u_t = k\frac{\partial}{\partial x}\frac{\partial u}{\partial x} \qquad 0<x<1;\ t>0; \tag{11}$$

Here k is constant. The finite difference formula of Eq.(11) is:

$$(2+\rho)\,u_j^{(n+1)} = u_{j-1}^{(n+1)} + u_{j+1}^{(n+1)} + \rho u_j^{(n)} \tag{12}$$

Here

$$\rho = \frac{h^2}{k\Delta t} \qquad j = 0,1,2,\ldots M;\ n = 0,1,2,\ldots$$

$u_0^{(n)} = 0 \quad u_M^{(n)} = 1$
$u_j^{(0)} = 0$

The h and Δt are step length of x and t respectively. The iterative formula of Eq.(12) is

$$(2+\rho)\,u_j^{(n+1)(s+1)} = u_{j-1}^{(n+1)(s)} + u_{j+1}^{(n+1)(s)} + \rho u_j^{(n)} \tag{13}$$
$$s = 1,2,\ldots$$

From paper [1], following theorem can be easily proven.
Theorem 6: The limit value of the sequence $\{u_n^{(s)}\}$ is $u_n^{(s^*)}$.

We rewrite Eq.(13) in the following form:

$$(2+\rho)\,u_j^{(n+1)(s+1)} = u_{j-1}^{(n+1)(s)} + u_{j+1}^{(n+1)(s)} + \rho u_j^{(n)(*)} \tag{14}$$
$$s=1,2,\ldots.$$

Theorem 7. If $u_j^{(n)(s)}$ satisfy Eq.(14), and $u_j^{(n)(0)} = 0$, $j>1$, $n<N$, then

(a). $u_j^{(n+1)(s)} - u_j^{(n)(s)} \geq 0$
(b). $u_{j+1}^{(n)(s)} - u_j^{(n)(s)} \geq 0$
(c). $u_j^{(n)(s+1)} - u_j^{(n)(s)} \geq 0$
(d). $u_{j+1}^{(n)(s)} - 2u_j^{(n)(s)} + u_{j-1}^{(n)(s)} \geq 0$ (15)

Proof: When $n = 1$, we can easily prove Eq.(14).
Suppose Eq.(14) is correct when n = k. That is

(a). $u_j^{(k+1)(s)} - u_j^{(k)(s)} \geq 0$
(b). $u_{j+1}^{(k)(s)} - u_j^{(k)(s)} \geq 0$
(c). $u_j^{(k)(s+1)} - u_j^{(k)(s)} \geq 0$
(d). $u_{j+1}^{(k)(s)} - 2u_j^{(k)(s)} + u_{j-1}^{(k)(s)} \geq 0$ (16)

When $n = k+1$ we need prove following equations

(a). $u_j^{(k+2)(s)} - u_j^{(k+1)(s)} \geq 0$
(b). $u_{j+1}^{(k+1)(s)} - u_j^{(k+1)(s)} \geq 0$
(c). $u_j^{(k+1)(s+1)} - u_j^{(k+1)(s)} \geq 0$
(d). $u_{j+1}^{(k+1)(s)} - 2u_j^{(k+1)(s)} + u_{j-1}^{(k+1)(s)} \geq 0$ (17)

When $s = 0$ we have

$$(2+\rho)\,u_j^{(k+1)(1)} = \rho u_j^{(k)(*)}$$
$$(2+\rho)\,u_j^{(k+2)(1)} = \rho u_j^{(k+1)(*)} \geqslant \geqslant \rho u_j^{(k)(*)} = (2+\rho)\,u_j^{(k+1)(1)}$$

or $u_j^{(k+2)(1)} \geqslant u_j^{(k+1)(1)}$

and

$$(2+\rho)\,(u_{j+1}^{(k+1)(1)} - u_j^{(k+1)(1)}) = \rho\,(u_{j+1}^{(k)(*)} - u_j^{(k)(*)}) \geqslant 0$$
$$(2+\rho)\,(u_j^{(k+1)(1)} - u_j^{(k+1)(0)}) = \rho\,(u_j^{(k)(*)} - 0) \geqslant 0$$
$$(2+\rho)(u_{j+1}^{(k+1)(0)} - 2u_j^{(k+1)(0)} + u_{j-1}^{(k+1)(0)} = \rho\,(u_{j+1}^{(k)(*)} - 2u_j^{(k)(*)} + u_{j-1}^{(k)(*)}) \geqslant 0$$
$$j = 1,2,\ldots,N-1$$

Thus it have been proven that Eq.(17) is correct when $s = 0$.
Suppose Eq.(17) is correct when $s = m$, or

(a). $u_j^{(k+2)(m)} - u_j^{(k+1)(m)} \geqslant 0$
(b). $u_{j+1}^{(k+1)(m)} - u_j^{(k+1)(m)} \geqslant 0$
(c). $u_j^{(k+1)(m+1)} - u_j^{(k+1)(m)} \geqslant 0$
(d). $u_{j+1}^{(k+1)(m)} - 2u_j^{(k+1)(m)} + u_{j-1}^{(k+1)(m)} \geqslant 0$ (18)

When $s = m+1$, from Eq.(17), we have

$$(2+\rho)\,(u_j^{(k+2)(m+1)} - u_j^{(k+1)(m+1)}) = [u_{j+1}^{(k+2)(m)} - u_{j+1}^{(k+1)(m)}] + [u_{j+1}^{(k+2)(m)} - u_{j+1}^{(k+1)(m)}] + \rho[u_j^{(k+1)(*)} - u_j^{(k)(*)}] \geqslant 0$$

$$(2+\rho)\,(u_{j+1}^{(k+1)(m+1)} - u_j^{(k+1)(m+1)}) = [u_{j+2}^{(k+1)(m)} - u_{j+1}^{(k+1)(m)}] + [u_j^{(k+1)(m)} - u_{j-1}^{(k+1)(m)}] + \rho[u_{j+1}^{(k+1)(*)} - u_j^{(k)(*)}] \geqslant 0$$

$$(2+\rho)\,(u_j^{(k+1)(m+2)} - u_j^{(k+1)(m+1)}) = [u_{j+1}^{(k+1)(m+1)} - u_{j+1}^{(k+1)(m)}] + [u_{j+1}^{(k+1)(m+1)} - u_{j+1}^{(k+1)(m)}] + \rho[u_j^{(k)(*)} - u_j^{(k)(*)}] \geqslant 0$$

$$(2+\rho)\,(u_{j+1}^{(k+1)(m+1)} - 2u_j^{(k+1)(m+1)} + u_{j-1}^{(k+1)(m+1)}) = [u_{j+2}^{(k+1)(m)} + u_j^{(k+1)(m)}] - 2[u_{j+1}^{(k+1)(m)} + u_{j-1}^{(k+1)(m)}] + [u_j^{(k+1)(m)} + u_{j-2}^{(k+1)(m)}] + \rho[u_{j+1}^{(k)(*)} - 2u_j^{(k)(*)} + u_{j-1}^{(k)(*)}] \geqslant (u_{j+2}^{(k+1)(m)} - 2u_{j+1}^{(k+1)(m)} + u_j^{(k+1)(m)}) + (u_{j-1}^{(k+1)(m)} - 2u_j^{(k+1)(m)} + u_{j-2}^{(k+1)(m)}) \geqslant 0$$

Therefore the Eq.(17) is also correct when $s = m+1$. #

Suppose the polynomial of degree 2 is

$$f^{(n)}(y) = u_j^{(n)(s)} + a(y-s) + b(y-s)^2 \tag{19}$$

and

$$f^{(n)}(s-1) = u_j^{(n)(s-1)}, \qquad f^{(n)}(s+1) = u_j^{(n)(s+1)}$$

We can easily prove the following theorems 8 and 9.

Theorem 8. $a = 0.5(u_j^{(n)(s+1)} - u_j^{(n)(s-1)})$
$b = 0.5(u_j^{(n)(s-1)} - 2u_j^{(n)(s)} + u_j^{(n)(s+1)})$

Theorem 9. Given fixed j and n,
for any $s>1$, and $u_j^{(n)(s+1)} - u_j^{(n)(s)} \geqslant 0$,
Eq.(9) has a maximal value $u_j^{(n)(*)} = u_j^{(n)(s)} - 0.25a^2/b$
if and only if $u_j^{(n)(s)}$ satisfies Eq.(15).

From above theorems, theorem 10 can be derived:

Theorem 10. Suppose $d_s \geqslant d_{s-1}$ and $d_{s+1} - 2d_s + d_s \geqslant 0$
$f(y) = d_s + a(y-s) + b(y-s)^2$
then $f(y)$ has a maximal value while
$d_s^{(*)} = d_s - 0.25a^2/b$
$a = 0.5(d_{s+1} - u_{s-1}) \qquad b = 0.5(d_{s-1} - 2d_s + u_{s+1})$
if and only if d_s satisfy following formula
$d_{s+1} - 2d_s + d_s \geqslant 0$

4. NUMERICAL RESULTS

Convergence of the VBF method so far has been proven with two examples. This means the VBF method is in deed an efficient approach in domain decomposition parallel computing. General proofs for different categorised Partial Differential Equation (PDE) solver with the VBF method are difficult. Some work, presented in other papers, has been done [7]. However intensive researches of the VBF method are still needed.

We have implemented two parallel numerical methods for the Example 2 on a local network: one using the VBF method and the other normal iteration method; both are programmed in the PVM platform on a local network.

In the implementations, iteration stopped when numerical values of function u_j are converged on whole domain. The convergence criteria is, $|U^{(s+1)} - U^{(s)}| < \varepsilon$ between two iterations.

Suppose η is a ratio of parallel calculation efficiency of our VBF method over normal iteration method in solving example 2, and P is number of computers used. The implementations

involve 50,000 nodes of heat flow with cyclic boundary conditions given in [5]. For error bound $\varepsilon = 0.0001$ and minimum number of iterations $N_{min} = 4P$ we can see from Table 1 that for any number of computers used, the ratio of parallel calculation efficiency is greater than 1.0. When $P < 3$, the $\eta \sim 1.0$, however when $P > 5$ the $\eta > 2.80$. This indicates that computers are busy in communication in the normal parallel iterative algorithm and the VBF method has less communication time and higher parallel extent.

Table 1. Parallel calculation efficiency ratio of the VBF method over normal iterative method

P	1	2	3	4	6	8	10
η	1	.025	1.35	2.10	3.26	4.11	5.42

5. CONCLUSIONS

This paper has proved necessary and sufficient conditions of convergence for the VBF algorithm using two examples. We have shown that the VBF method could reduce communication dramatically. The theorems 1 to 5 are basic VBF theorems, which proved that static equation satisfies the condition of VBF method. The theorems 6 -10 are used for solving transient equation with VBF method, which proved that linear transient equation satisfies the condition of VBF method. For non-linear transient equation and other equations correctness and efficiency of the VBF method are still remained to prove. In our examples the numerical results show that the VBF method can bring high efficiency for parallel calculations.

REFERENCES

[1].Wang xianfu & Zhang Shesheng, Hydrofoil supercave flow with free surface. Jour. of Shipbuilding, No.4 (1996). pp1-8.

[2]. Guo Qingping , Y. Paker, et al, Optimum Tactics of Parallel Multi-grid Algorithm with Virtual Boundary Forecast Method Running on a Local Network with the PVM Platform, Journal of Computer Science and Technology, July 2000, Vol.15, No.4, pp355~359

[3].Guo Qingping & Yakup Paker, Concurrent Communication and Granularity Assessment for a Transputer-based Multi-processor System, Journal of Computer Systems Science & Engineering,Vol.5 No.1.January,1990.

[4]. L.R. Matheson et al, Parallelism in Multigrid Methods: How Much is too Much, International Journal of Parallel Programming, Vol.24, No.5, Plenum Publishing Corporation, 1996.

[5]. 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.

[6] Fan Yingquan. Advance mathematics. 1962, Beijing. pp276~312.

[7]. Wei Jianing, Guo Qingping et al., A Multigrid Parallel Algorithm of One Dimensional Virtual Boundary Ransack Forecast, Journal of Wuhan Transportation University, Wuhan China, Vol. 24 No. 2, April 2000, pp108~112

The IBiCG Method for Large and Sparse Unsymmetric Linear Systems on BSP Architectures

Laurence Tianruo Yang
Department of Computer Science, St. Francis Xavier University
P.O. Box 5000, Antigonish, B2G 2W5, Nova Scotia, Canada

Xiangzhen Qiao
National Research Center for Intelligent Computing Systems (NCIC)
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, P. R. China 100080

Ruth E. Shaw
Department of Computer Science, University of New Brunswick
Saint John,E2L 4L5, New Brunswick, Canada

ABSTRACT

For the solutions of large and sparse linear systems of equations with unsymmetrical coefficient matrices, we propose an improved version of the BiConjugate Gradient method (IBiCG) method based on [2,3] by using the Lanczos process as a major component combining elements of numerical stability and parallel algorithm design. The algorithm is derived such that all inner products, matrix-vector multiplications and vector updates of a single iteration step are independent and communication time required for inner product can be overlapped efficiently with computation time of vector updates. Therefore, the cost of global communication on parallel-distributed memory computers can be significantly reduced. In this paper, we use the Bulk Synchronous Parallel (BSP) model to design a fully efficient, scalable and portable parallel IBiCG algorithm and to provide accurate performance prediction of the algorithm for a wide range of architectures including the Cray T3D, the Parsytec GC, and a cluster of workstations connected by an Ethernet. This performance model provides us useful insight in the time complexity of the IBiCG method using only a few system dependent parameters based on a simple and accurate cost modeling. The theoretical performance prediction is compared with measured timing results of a numerical application from ocean flow simulation.

Keywords: The BiConjugate Gradient (BiCG) method, Unsymmetrical linear systems, Lanczos process, Bulk synchronous parallel (BSP) model, Performance analysis.

1. INTRODUCTION

The solution of linear systems with unsymmetrical coefficients arises frequently in scientific computing, for example from finite difference or finite element approximations to partial differential equations, as intermediate steps in computing the solution of nonlinear problems or as sub problems in linear and nonlinear programming.

In this paper, we will mainly focus on one Krylov subspace method, namely, the Biconjugate gradient algorithm (BiCG) [9, 8], for large and sparse linear systems with unsymmetrical coefficient matrices.

The author's work is supported by NSERC of Canada

The basic time-consuming computational kernels of BiCG are usually: inner products, vector updates and matrix-vector multiplications. In many situations, especially when matrix operations are well structured, these operations can be efficiently implemented on vector and shared memory parallel computers [7]. But for parallel distributed memory machines, the matrices and vectors are distributed over the processors, so that even when the matrix operations can be implemented efficiently by parallel operations, we still cannot avoid the global communication, i.e. communication involving all processors, required for inner product computations. Vector updates are perfectly parallelizable and, for large sparse matrices, matrix vector multiplications can be implemented with communication between only nearby processors. The bottleneck is usually due to inner products enforcing global communication. The detailed discussions on the communication problem on distributed memory systems can be found in [4, 6]. These global communication costs become relatively more important when the number of the parallel processors increases, subsequently they have the potential to affect the scalability of the algorithm in a negative way [4, 6].

Recently, Biiucker et al. [2, 3] propose a new modified parallel version of the BiConjugate Gradient (MbiCG) method. The algorithm is derived that both generated sequences of Lanczos vectors are scalable and is reorganized without changing the numerical stability so that there is only a single global synchronization point per iteration. Based on their similar ideas, we propose a new improved two-term recurrences Lanczos process without lookahead as the underlying process for the new Improved BiConjugate Gradient (IBiCG) method. The algorithm is reorganized without changing the numerical stability so that all inner products, matrix-vector multiplications and vector updates of a single iteration step are independent, and subsequently communication time required for inner product can be overlapped efficiently with computation time of vector updates. Therefore, the cost of global communication on parallel-distributed memory computers can be significantly reduced. The resulting IBiCG algorithm maintains the favorable properties of the Lanczos process while not increasing computational costs.

However, the performance model in [17] of computation and communications phases based on [5, 15] presented to give a quantitative analysis of the parallel performance is still very limited and is hard to achieve portability due to the underlying

assumption of a special network architecture. The Bulk Synchronous Parallel (BSP) model provides software developer with an attractive escape route from the world of architecture dependent parallel algorithms. A rapidly growing community of software and algorithm designers who are eager to produce scalable and portable parallel software and algorithms has enthusiastically welcomed the BSP. We use it to design a fully efficient, scalable and portable parallel IBiCG algorithm and to provide accurate performance prediction. This performance model provides us useful insight in the time complexity of the IBiCG method using only a few system dependent parameters based on a simple and accurate cost modeling.

The theoretical results in the performance is verified by measured timing results from an ocean flow simulation problem carried out on different parallel architectures: the Cray T3D, the Parsytec and a cluster of SUN workstations connected by an Ethernet (all three can be regarded as a BSP machine).

The paper is organized as follows. In section2, The Improved Biconjugate Gradient (IBiCG) method is described shortly. In section 3, parallel performance BSP model to be used in the analysis is outlined including the communication model and assumptions for computation time and communication costs. Finally parallel performance is discussed in two ways, namely theoretical complexity analysis and experimental observations in sections 4 and 5, respectively.

2. THE IBiCG METHOD

The improved Lanczos process is used as a major component to a Krylov subspace method for solving a system of linear equations

$$Ax = b, \text{ where } A \in \Re^{n\times n} \quad x, b \in \Re^n. \quad (1)$$

In each step, it produces approximation x_n to the exact solution of the form

$$x_n = x_0 + K_n(r_0, A), n = 1,2,... \quad (2)$$

Here x_0 is any initial guess for the solution of linear systems, $r_0 = b - Ax_0$ is the initial residual, and

$$K_n(r_0; A) = span\{r_0, Ar_0, \cdots, A^{n-1}r_0\},$$

is the n-th Krylov subspace with respect to r_0 and A.

Given any initial guess x_0, the n-th iterate is of the form

$$x_n = x_0 + V_n z_n, \quad (3)$$

where V_n is generated by the improved unsymmetrical Lanczos process [2, 3], and z_n is determined by the property.

Under the assumptions, the Improved BiConjugate gradient (IBiCG) method using improved Lanczos process as underlying process can be efficiently parallelized as follows:

- The inner products of a single iteration step (16), (17), (18), and (19) are independent.
- The matrix-vector multiplications of a single iteration step (11) and (12) are independent.
- The vector updates (13), (14) and (15) are independent. The vector updates (9) and (10) are independent.
- The communications required for the inner products (16), (17), (18) and (19) can be overlapped with the update for p_n in (20).
- Therefore, the cost of communication time on parallel distributed memory computers can be significantly reduced.

Algorithm 1 The IbiCG Method

1: $p_0 = q_0 = 0, v_1 = w_1 = b - Ax_0$
2: $\gamma_0 = \xi_0 = 0, \tau_0 = \rho_0 \neq 0, \kappa_0 = -1$
3: $a_0 = A^T q_0, b_0 = Ap_0, s_0 = A^T w_1$
4: $\gamma_1 = \|v_1\|, \xi_1 = \|w_1\|, \rho_1 = w_1^T v_1, \varepsilon_1 = s_1^T v_1$
5: for n=1,2,3.....do
6: $\mu_n = \frac{\gamma_{n-1}\varepsilon_{n-1}\rho_n}{\gamma_n\tau_{n-1}\rho_{n-1}}$
7: $\tau_n = \frac{\varepsilon_n}{\rho_n} - \gamma_n\mu_n$
8: $\kappa_n = -\frac{\gamma_n}{\tau_n}\kappa_{n-1}$
9: $p_n = \frac{1}{\gamma_n}v_n - \mu_n p_{n-1}$
10: $q_n = \frac{1}{\xi_n}s_n - \frac{\gamma_n\mu_n}{\xi_n}q_{n-1}$
11: $a_n = A^T q_n$
12: $b_n = Ap_n$
13: $s_{n+1} = a_n - \frac{\tau_n}{\xi_n}s_n$
14: $v_{n+1} = b_n - \frac{\tau_n}{\gamma_n}v_n$
15: $w_{n+1} = q_n - \frac{\tau_n}{\xi_n}w_n$
16: $\gamma_{n+1} = v_{n+1}^T v_{n+1}$
17: $\xi_{n+1} = w_{n+1}^T w_{n+1}$
18: $\rho_{n+1} = w_{n+1}^T v_{n+1}$
19: $\varepsilon_{n+1} = s_n^T v_{n+1}$
20: $x_n = x_{n-1} + \kappa_n p_n$
21: *if* $(|\gamma_{n+1}\kappa_n| < tol)$ *then*
22: STOP
23:end if
24:end for

3. THE BSP ARCHITECTURE

The BSP model was proposed by Valiant [12]. It defines architecture, a type of algorithm, and a function for charging costs to algorithms. Our working platforms are selected from massively distributed memory computers such as the Cray T3D, the Parsytec GC, and a cluster of SUN workstations connected via Ethernet that all can be regarded as BSP

architectures. Here we use the variant of the cost function proposed in [1]. For a detailed survey of the BSP architecture, see [10].

Generally, a BSP architecture can be characterized by four parameters as follows

- p is the number of processors.
- s is the single processor speed measured in flop/s.
- g is the communication cost per data word.
- l is the synchronization cost of a superstep.

Accordingly, the execution time of an algorithm with cost $a+bg+d$ on a BSP architecture with four parameters (p; s; g; l) is $(a+bg+d)/s$ seconds. For numerical timing experiments, we use two parallel distributed memory computers, CrayT3D and Parsytec GC, and workstation clusters of SUN workstations which are connected via Ethernet with a maximal throughput of 10 Mbit/s. The nodes of the CrayT3D used consist of a 150 MHz Dec Alpha processor and a memory of 64 Mbytes. Both communication networks of CrayT3D and Parsytec GC are three-dimensional torus. All experimental machines are transformed into BSP architectures by using the corresponding implementations of the Oxford BSP library BSPL- IB [11]. Usually the different sequential computing speeds s for different machines are obtained by measuring the time of a vector update operation.

Values of the BSP cost model parameters with 32 bits reals as data word are shown in Table 1. From the table, not very surprisingly, the two values of g, derived directly from a 1-relation, and from a h-relation total exchange are different. This might mean that the 1-relation performance of the network is not very good, but usually means that the network's effective capacity is not as large as the per link bandwidth would suggest. When cost modeling Algorithms, it is advisable to use the value of g produced by the total exchange benchmark. Hereafter we will use the value of g measured by the total exchange benchmark for the cost function. When p = 1,g represents the memory speed of the processor, taking into account any buffering communication that may occur in the implementation of BSPL-IB. The efficiency of the communication network can also be roughly estimated by comparing the value of g for one processor with the value of g for p > 1. This gives the ratio of the memory speed to the interprocessor communication speed.

Table 1 shows that the processor speed is independent ofthenumber of processors. For the parameters g and l we make a crude simplification by assuming a linear dependence on the number of processors. We get the following linear least squares curves suggested by [13, 14]of g and l for different parallel machines respectively. For CrayT3D, we have

$$\begin{aligned} l_{CrayT3D} &= 0.45p+149 \\ g_{CrayT3D} &= 0.016p+0.77 \end{aligned} \quad (4)$$

For Parsytec GC, we have

$$\begin{aligned} l_{Parsytec} &= 14630p-33065 \\ g_{Parsytec} &= 20.23p+45.63 \end{aligned} \quad (5)$$

For SUN-workstation clusters, we have

$$l_{Clusters} = 31p-6, g_{Clusters} = 1.08p+0.15 \quad (6)$$

4. THEORETICAL ANALYSIS

Our numerical example is a very simple model from [14] for the spreading of pollution from a small source in the Pacific. This model can be modeled by the Possion equation with proper right side and boundary conditions. The problem is formulated based on [13, 14] in spherical coordinates, with constant radius, and the results are expressed in the familiar longitudes and latitudes.

Due to the special matrix structure described previously and background in ocean flow simulation, we use domain decomposition to parallelize the matrix-vector multiplication as well as the inner product and vector update operation described in [13, 14]. For the general sparse cases, the described approach in [1] strongly recommended. In the domain decomposition approach all elements are uniquely

Table 1: Bulk Synchronous Parallel Machine Parameters

Machine	p	s	l		g(1-relation)		g(h-relation)		$n_{1/2}$ words
	M flops/s		flops	us	flop/word	us/word	flop/word	us/word	
CrayT3D	12	1	64	5.6	0.4	0.02	0.3	0.18	88
		2	160	13.4	0.7	0.06	1.0	0.07	73
		4	167	13.7	0.7	0.06	1.1	0.68	68
		8	176	14.4	0.8	0.06	0.9	0.68	58
		16	184	15.2	0.9	0.07	1.0	0.07	63
Parsytec GC	19	1	99	5.2	1.1	0.06	1.1	0.06	18
		2	6309	326	109	5.7	113	6.0	4
		4	23530	1220	100	10.6	145	7.8	4
		8	29080	1506	250	13.6	254	13.4	4
Workstation Clusters	10	1	25	2.5	0.5	0.05	0.5	0.05	8
		2	56	5.4	2.8	0.29	3.4	0.33	8
		4	118	11.7	3.7	0.36	4.1	0.40	8

Table 2: Computations and communication per iteration

	Matrix-vector	Inner Product	Vector update
h	nb	$2p$-2	0
supersteps	2	4	0
ω	$14n$	$2n$	$2n$
frequency per iteration without overlapping	1	1	6
frequency per iteration with overlapping	1	1	5

assigned to a subdomain. The subdomains share grid points at the boundaries, and overlap in this sense. The domain decomposition and subsequent parallelization are described in some details in [13, 14].

In domain decomposition approach, the matrix-vector multiplication comprises one computation and one communication step. The number of floating point operations w is 14n with n being the number of grid points per subdomain. The number of reals to be sent and received, h is equal to the nonzero elements, nb, with nb being the number of boundary nodes. The inner product is composed of two computation and two communication steps. Only p_1 reals, with p being the number of processors, are received in the first communication step, and p_1 reals are sent in the second communication step. In the two computation steps w = 2n flops are performed on each processor. The vector update operation requires 2n ops and no communication. The communications and computations per iteration of various operations are tabulated in Table 2. Table 2 also gives the number of matrix-vector multiplication, inner products and vector update in an IBiCG iteration. From Table 2 a total cost function for an IBiCG iteration can be determined. For the case of overlapping between communication and computation, the vector update operation can be efficiently overlapped with the communication of inner product operation described in the IBiCG method. Here since there are two communication and two computation steps during an inner product operation, we can perform the computations of the vector update operation with the computation step of inner product operation simultaneously. Hence the total number of super steps per iteration is 6. The total number of reals to be fetched or stored per iteration is $h = nb + 2p$-2. The total number of floating point operations per iteration is w = 46n in the overlapping case, compared to w = 48n in the case without overlapping. So there is no significant improvement in the total cost using the overlapping technique. In the discussions throughout the rest of the paper we will use the cost function of the overlapping case and the same methodology used in [13, 14]. The total cost function per iteration of the IBiCG method is thus

$$T_{cost} = 46n + (nb + 2p - 2)g + 6l \qquad (7)$$

The function Tcost gives workload in op. The first term corresponds to computation, the second to communication, and the third to synchronization cost. For the prediction of the actual running time we have to divide T_{cost} by the processor speed s described previously.

Parameters n and nb depend on the problem and on the domain decomposition. For the test problem we have the number of boundary points nb = 90,and the number of grid points per domain n = 45(1+90/p). Parameters s, g and l are system dependent which have been described in section 4. Combining the relations (4), (5) and (6) we obtain the following cost functions per iteration for the CrayT3D, the Parsytec GC and the SUN-workstation clusters respectively.

For the CrayT3D, the cost function per iteration is

$$T_{cost}^{CrayT3d} = \frac{186}{p} + 3.0 + 5.7*10^{-3}p + 3.2*10^{-5}p^2,$$

which is measured in K flops. Correspondingly, the theoretical prediction of the actual running time per iteration is given as follows:

$$T_{runtime}^{CrayT3D} = \frac{15.5}{p} + 0.25 + 4.8*10^{-4}p + 2.7*10^{-6}p^2$$

which is measured in ms. As can be noted that when p = 118, the cost function per iteration is minimal and equals 5:76Kflops, hence the minimal running time is 0:48ms. If the number of processors is less than p = 118, the running time decreases when the number of processors increases. If the number of processors is larger than p = 118,we can see that the running time increases as the number of processors increases.

For the Parsytec, the cost function per iteration is

$$T_{cost}^{PrayT3d} = \frac{186}{p} - 198 + 90p + 0.04p^2$$

which is measured in K ops. Correspondingly, the theoretical prediction of the actual running time per iteration is given as follows:

$$T_{runtime}^{PrayT3D} = \frac{9.8}{p} - 10.4 + 4.7p + 2.1*10^{-3}p^2$$

which is measured in ms. From the above equation it can be shown that for p >3, the execution time increases if the number of processors increases.

For SUN-workstation clusters, the cost function per iteration is

$$T_{cost}^{clusters} = \frac{186}{p} + 2 + 0.28p + 2*10^{-3}p^2$$

which is measured in K ops. Correspondingly, the theoretical

Table 3: Predicted and measured time per iteration on the CrayT3D

# of proc.	Predicted (ms)				Measured (ms)
	Tromp	T_{comm}	T_{sync}	T_{total}	T_{total}
4	4.054	0.007	0.075	4.136	4.127
8	2.113	0.008	0.076	2.197	2.172
16	1.143	0.010	0.078	1.231	1.237
32	0.658	0.016	0.082	0.756	0.752
64	0.415	0.032	0.089	0.536	0.523
118	0.304	0.072	0.101	0.477	0.474
128	0.294	0.081	0.103	0.479	0.486
256	0.233	0.243	0.132	0.609	0.617

Prediction of the actual running time per iteration is given as follows:

$$T_{runtime}^{Clusters} = \frac{19}{p} + 0.2 + 0.028p + 2*10^{-4}p^2$$ Which is measured in ms. For p = 25, the cost function per iteration is at

a minimum of 17:9K flops, and the minimal running time is 1:79ms. For $p \leq 25$, the running time decreases when the number of processors Increases. But the running time increases as the number of processors becomes larger than 25.

5. EXPERIMENTAL RESULTS

In this section, we present the results of several numerical timing experiments to compare with the predictions of our performance model. In Table 3, Table 4 and Table 5, the predicted and measured running time per iteration of the IBiCG method on the CrayT3D, the Parsytec GC and SUN-workstation clusters are given respectively. On the CrayT3D, there is no significant communication overhead for p <118. The corresponding running time per iteration decreases as the number of processors is increased just as predicted by our theoretical analysis. For p = 118, we achieve the minimal running time 0:48ms, and optimal speedup. Increasing p further from 118, the communication becomes the bottleneck and dominates the running time, which leads to an increasing running time. The predicted and measured running time is very close to each other, although the predicted times are continuously slightly higher than the measured run time. The quantitative behavior is described and predicted quite very well by our analysis.

For the Parsytec GC and SUN-workstation clusters, we obtain similar conclusions as for the CrayT3D. It is clear that for increasing number.

Table 4: Predicted and measured time per iteration on the Parsytec GC

# of proc.	Predicted (ms)				Measured
	Tcomp	T_{comm}	T_{sync}	T_{total}	T_{total}
3	3.377	0.526	3.143	7.046	7.124
4	2.560	0.639	7.766	10.965	11.144
8	1.335	1.136	26.257	28.727	28.506
16	0.722	2.333	63.239	66.294	66.307
32	0.415	5.544	137.205	143.164	143.114
64	0.262	15.238	285.136	300.636	300.502

of processors, the computation time per iteration decreases. The communication costs per iteration become more expensive for a larger number of processors. Thus they have the potential to affect the scalability of the algorithm in a negative way. This conclusion is consistent with the results [15, 16, 17] on distributed memory computers. The fact that the predicted run time is higher than the measured run time can be partly explained by the way the benchmark is carried out. Timing the vector update operations mainly derives the processor speed used in the performance model. But for example an innerproduct operation can be performed more efficiently. Accordingly, the processor speed is estimated too low for the theoretical analysis. The synchronization parameter l is estimated for messages of length larger than 1. The inner product operation requires communication with messages of length 1. Hence, l is estimated also too high for our theoretical analysis. The difference between the predicted and measured times is small and our performance model describes the qualitative behavior of the IBiCG method on BSP architectures quite well.

Table 5: Predicted and measured time per iteration on SUN-workstations

# of proc.	Predicted(ms)				Measured(ms)
	Tromp	T_{comm}	T_{sync}	T_{total}	T_{total}
2	9.522	0.021	0.0349.	9.622	2
4	4.865	0.043	0.0715.	4.865	4
8	2.536	0.091	0.1452.	2.761	8
16	1.371	0.209	0.2941.	1.762	16
25	0.952	0.375	0.4611.	1.769	25
32	0.789	0.528	0.5921.	1.882	32

REFERENCES

[1] R. H. Bisseling and W. F. McColl. Scientific computing on bulk synchronous parallel architectures. Technical Report TR-836, Department of Mathematics, Utrecht University, December 1993.

[2] H. M. Bucker and M. Sauren. A Variant of the Biconjugate Gradient Method Suitable for Massively Parallel Computing. In G. Bilardi, A. Ferreira, R. Luling, and J. Rolim, editors, *Solving Irregularly Structured Problems in Parallel, Proedings of the Fourth International Symposium, IRREGULAR'97*, Paderborn, Germany, June 12{13, 1997, volume 1253 of *Lecture Notes in Computer Science*, pages 72{79, Berlin, 1997. Springer.

[3] H. M. Bucker and M. Sauren. Parallel biconjugate gradient methods for linear systems. In L. T. Yang, editor, *Parallel Numerical Computations with Applications*, number 51-70. Kluwer Academic Publishers, 1999.

[4] E. de Sturler. A parallel variant of the GMRES(m). In *Proceedings of the 13th IMACS World Congress on Computational and Applied Mathematics*. IMACS, Criterion Press, 1991.

[5] E. de Sturler. Performance model for Krylov subspace methods on mesh-based parallel computers. Technical Report CSCS-TR-94-05, Swiss Scientific Computing Center, La Galleria, CH-6928 Manno, Switzerland, May 1994.

[6] E. de Sturler and H. A. van der Vorst. Reducing the effect of the global communication in GMRES(m) and CG on parallel distributed memory computers. Technical Report 832, Mathematical Institute, University of Utrecht, Utrecht, The Netherland, 1994.

[7] J. J. Dongarra, I. S. Duff, D. C. Sorensen, and H. A. van der Vorst. *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, PA, 1991.

[8] R. Fletcher. Conjugate Gradient Methods for Indefinite Systems. In G. A. Watson, editor, *Numerical Analysis Dundee 1975*, volume 506 of *Lecture Notes in Mathematics*, pages 73{89, Berlin, 1976. Springer.

[9] C. Lanczos. Solutions of Systems of Linear Equations by Minimized Iterations. *Journal of Research of the National Bureau of Standards*, 49(1):33{53, 1952.

[10] W. F. McColl. Scalable computing. In J. van Leeuwen, editor, *Computer Science Today: Recent Trends and Developments*, volume 1000 of *Lecture Notes in Computer Science*, pages 46{61. Springer Verlag, Berlin, 1995.

[11] R. Miller and J. Reed. *The Oxford BSP library users'*

guide. Oxford Parallel, Oxford University, 1st edition, 1993.

[12] L. G. Valiant. A bridging model for parallel *Computation. Communication of ACM*, 33(8):103{111, 1990.

[13] M. B. van Gijzen. Parallel iterative solution methods for linear fonite element computations on the CrayT3D. *In Proceedings of the High Performance Computing and Networking 1995*, number 919 in Lecture Notes in Computer Science, pages 723{728. Springer Verlag, 1995.

[14] M. B. van Gijzen. Parallel ocean flow computations on a regular and on an irregular grid. In *Proceedings of the High Performance Computing and Networking 1996*, number 1067 in Lecture Notes in Computer Science, pages 207{212. Springer Verlag, 1996.

[15] T. Yang. Solving sparse least squares problems on massively parallel-distributed memory computers. In Proceedings of *International Conference on Advances in Parallel and Distributed Computing (APDC-97)*, March 1997. Shanghai, P.R.China.

[16] T. Yang and H. X. Lin. The improved quasi-minimal residual method on massively distributed memory computers. In Proceedings of *The International Conference on High Performance Computing and Networking (HPCN-97)*, April,Vienna, Austria 1997.

[17] T. Yang and H. X. Lin. Performance prediction of the parallel improved quasi-minimal residual method for large and sparse unsymmetrical linear systems. In Proceedings of *9th International Conference on Parallel and Distributed Computing and Systems (PDCS-97)*, October 1997. Washington, U.S.A.

Parallel Modular Arithmetic Based on Signed-Digit Number System and the Application to Error Detection of Product-Sum Computation

Shugang Wei Kensuke Shimizu
Department ofComputer Science, Gunma University,
1-5-1, Tenjin-cho, Kiryu, 376-8515 Japan
wei@cs.gunma-u.ac.jp

ABSTRACT

Parallel modulo m (m=2p±1) adder and multiplier an d binary-to-residue number converter are presented by using a radix-2 signed-digit (SD) number representation. The proposed arithmetic circuits can be used in a residue number system (RNS) for fast parallel/distributed computation. The modulo m addition is implemented by using a p-digit SD adder, so that the modulo m addition time is independent of the word length of operands. Thus, with a binary tree structure of the SD modulo m adders, the modulo m multiplication is performed in a time proportional to log 2 p and an n-bit binary number is converted into a p-digit SD residue number in a time proportional to log 2 (n=p). The presented modular arithmetic circuits can be applied to error detection for an large product-sum circuit.

Keywords: Residue Number System (RNS), Modular Addition, Modular Multiplication, Signed-digit (SD) Number Representation, SD Adder, Error Detection.

1. INTRODUCTION

High speed computations are general requirements in real-time applications such as high speed digital sign- al processing and digital control systems. The residue number system (RNS) has the well-known property that the *i*th residue digit of sum, difference, and product is exclusively dependent on the ith digit of the operands [1]. This property determines that truly parallel operations can be performed on all residue digits as a distributed computation system. Various methods of applications of RNS in digital signal processing have been proposed [2].

A modulo 2^p or $2^{p\pm1}$ addition can be implemented by a p-bit binary adder [1, 3]. Some modulo $2^{p\pm1}$ multipliers have been proposed for a residue number system (RNS)[4, 5]. By adding redundant module to an RNS, the error detection and error correct- ion can be done while performing arithmetic operations. Moreover, a checker with modular arithmetic can be also applied to detect the calculation error of an ordinary binary arithmetic circuit [6, 7]. In the conventional implementation of modular arithmetic, however, the ordinary binary arithmetic is performed, s o that the carry propagation will arise during additions and will limit the speed of arithmetic operations.

It is known that carry propagation is limited to one position during additions of signed-digit (SD) numbers [8]. To perform the high speed modular arithmetic, we have proposed a concept on modular arithmetic with the redundant residue representation using a p- digit radix-2 SD number system [9], in which the carry propagation is limited to one position during modular additions and the modular operation is easy to be performed. Based on the concept, in this paper, we present a fast modulo m ($(m \in \{2^p-1, 2^p, 2^p+1\})$. multiplier and a binary-to-residue converter with a binary tree structure of SD adders. Thus, the modulo m multiplication and the conversion from an n-bit binary to a p-digit SD residue numbers are performed in a time proportional to log 2 p and log 2 (n=p), respectively. It is considered that the presented modular arithmetic circuits can be used to detect the calculation error of a large product-sum circuit. We also give the design results of the presented circuits by using VH- DL.

2 REDUCUNDANT RESIDUE NUMBER

Set and Signed-Digit Number Representation In general, a residue digit $X=|X|_m = X-[X/M]\times M$ in a symmetric RNS has a value in the following number set:

$$l_m = \{-(m-1)/2, \ldots, (m-1)/2\}. \quad (1)$$

Where [X/M] is the closest integer rounding X/M In this paper, let $m \in \{2^p-1, 2^p, 2^p+1\}$.

Since the carry propagation arisen during additions in the ordinary binary arithmetic systems will limit the speed of arithmetic operations, we consider a residue number x represented by a *p*-digit radix-2 SD number representation as follows:

$$x = x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + \ldots + x_0, \quad (2)$$

$$x_i \in \{-1,0,1\} (i = 0,1,\ldots, p-1) \quad (3)$$

which can be denoted as $x = (x_{p-1}, x_{p-2}, \ldots, x_0)_{SD}$ In the SD number representation, x has a value in the range of $[-(2^p-1), 2^p-1]$ However, it is difficult to judge if x is in L_m.

To simplify the manipulation of the modular operation in SD number representation, the following redundant residue set is used.

$$L_m = \{-(2^p-1), \ldots, -1, 0, 1, \ldots, 2^p-1$$

Thus, x must be in L_m when it is expressed in a *p*-digit SD number representation. Obviously,

$$-X = -(x_{p-1}, x_{p-2}, \ldots, x_0)_{SD} = (-x_{p-1}, -x_{p-2}, \ldots -x_0)_{SD} \quad (4)$$

is in L_m.

Let X be an integer and $m \in \{2^p - 1, 2^p, 2^p + 1\}$ be a modulus.

then $x = (X)_m$ is defined as an integer in L_m. When $|X|_m \neq 0$ x has one of two possible values given by equations

$$X = (X)_m = |X|_m \quad (5)$$

and

$$X = (X)_m = |X|_m - \text{sign}(|X|_m) \times m \quad (6)$$

respectively, where

$$\text{sign}(x) = \begin{cases} -1 & x<0 \\ 1 & x \geq 0 \end{cases}$$

When $|X|_m = 0$, in the case of $m = 2^p - 1$ there are three possible values for x, that is, - m, 0 and m. The integer set lm in Eq.(1) is a partial set of Lm. The numbers as intermediate results calculated in Lm are used for some application with fast modular arithmetic. To obtain final results, if necessary, they can be converted into lm.

The redundant modular arithmetic has the following properties:

Property 1: Let a and b be integers. Then

(a) $\text{abs}((a)_m) \leq 2^p - 1$

(b) $(a \pm b)_m = ((a)_m \pm (b)_m)_m$

(c) $(a \times b)_m = ((a)_m \times (b)_m)$

and

(d) $(-a)_m = -(a)_m$

where abs(x) is the absolute value of x and $\equiv$ indicates a binary congruent relation with modulo m. s

3 Modular Arithmetic Circuits Based on the SD Adder

3.1 Modulo m SD adder

Figure 1 shows a modulo m SD adder (MSDA) having p SD full adders (SDFAs). Each SDFA is constructed with blocks ADD1 and ADD2, which have the following functions:

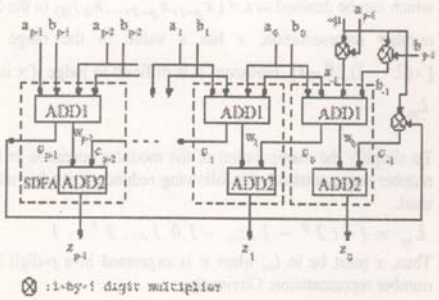


Figure 1: Modulo m signed-digit adder(MSDA)

(ADD1):

When $\text{abs}(a_i) = \text{abs}(b_i)$,

$$w_i = 0 \quad (7)$$

and

$$c_i = (a_i + b_i) \text{ div } 2; \quad (8)$$

When $\text{abs}(a_i) \neq \text{abs}(b_i)$,

$$w_i = \begin{cases} -(a_i + b_i) & \text{if } (a_i + b_i) \text{ and } (a_{i-1} + b_{i-1}) \\ & \text{have the same sign} \\ a_i + b_i & \text{otherwise} \end{cases} \quad (9)$$

and

$$c_i = \begin{cases} a_i + b_i & \text{if } (a_i + b_i) \text{ and } (a_{i-1} + b_{i-1}) \\ & \text{have the same sign} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

(ADD2): $Z_i = w_i + c_{i-1}$.

It is always true that $2ci + w_i = a_i + b_i$, and wi and c_{i-1} do not have the same sign so that $z_i \in \{-1,0,1\}$. Thus the carry propagation is limited to one digit and parallel arithmetic can be achieved without the carry propagation, which occurs during addition in an ordinary binary system.

Let $m = 2^p + \mu$ and $\mu \in \{-1,0,1\}$,

then $(2^p)_m = -\mu$ (11)

Thus,

$(c_{p-1} 2^p)_m = \mu . \times c_{p-1} = c_{-1}$

	i :	4	3	2	1	0
	a :	1	1	0	-1	1
(ADD1)	b :	0	1	-1	1	0
	w :		-1	0	-1	0	1
	c :	1	1	0	0	0	-1
(ADD2)							
	z :		0	0	-1	0	0

Figure 2: An example of modulo 33 SD addition

where $c_{-1} \in \{-1,0,1\}$, c_D and w_D are decided by using $-a_{p-1}\mu$ and $-b_{p-1}\mu$ as a_1 and a_1, respectively. Therefore, the modulo m addition time, $m = 2^p$ or $m = 2^p \pm 1$, constructed by an SD adder as shown in Fig.1 is independent of the word length. Figure 2 illustrates an example of modulo m SD addition, in which p = 5 and m =33. The calculation result is $(23 + 6)_{33} = -4$

3.2 Modulo m multiplier

Let x and y be two integers in the p-digit radix-2 SD number representation. Then

$x \times y = (x_{p-1} 2^{p-1} + x_{p-2} 2^{p-2} + \ldots\ldots + x_0)$

$$\times (y_{p-1}2^{p-1} + y_{p-2}2^{p-2} + ... + y_0)$$
$$= \sum_{i=0}^{p-1} y_i 2^i \times (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + ... + ...)$$

Thus, we have

$$(x \times y)_m = (\sum_{i=0}^{p-1} (y_i 2^i \times (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + ... + x_0))_m)_m$$
$$= (\sum_{i=0}^{p-1} pp_i)_m \quad (12)$$

Where

$$pp_i = (y_i 2^i \times (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + ... + x_0))_m \quad (13)$$

denotes as a partial product. Because of $y_i \in \{-1,0,1\}$ and replacing $(2^p)m$ with-u in the.

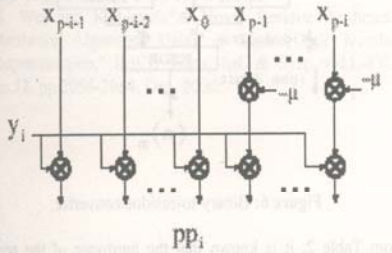


Figure 3: Partial product generation circuit(PPG)

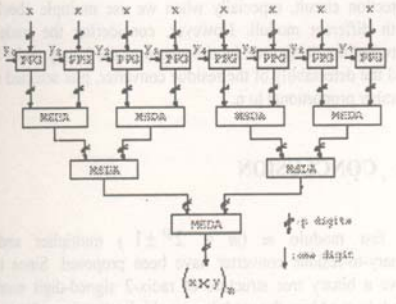


Figure 4: Modulo m multiplier

above equation, we have

$$pp_i = y_i(2^i \times (x_{p-1}2^{p-1} + x_{p-2}2^{p-2} + ... + x_0))_m$$
$$= y_i(2^p(x_{p-1}2^{i-1} + x_{p-2} + ... + x_{p-i})$$
$$+ x_{p-i-1}2^{p-1} + x_{p-i-2}2^{p-2} + ... + x_0 2^i)_m$$
$$= (x_{p-i-1}2^{p-1} + x_{p-i-2}2^{p-2} + ... + x_0 2^i$$
$$- -\mu(x_{p-1}2^{i-1} + ... + x_{p-i+1}2 + x_{p-i}))_m$$
$$= y_i \times (x_{p-i-1}, x_{p-i-2}, ... x_0,$$
$$-\mu x_{p-1}, ... -\mu x_{p-i+1}, -\mu x_{p-i})_{SD}$$

Therefore, a partial product is simply obtained by an i-digit end-around-shift and a p-by-1 digit multiplication, which can be performed in a constant time. Figure 3 shows a partial product generation circuit. A binary tree of the modulo m SD adders (MSDAs) can be constructed for the modulo m sum of the partial products as shown in Fig.4. The modulo m multiplication is performed in a time proportional to $\log_2^p$

Table 1: performance of modulo m multipliers

	Number of Gates	delay ime(ns)
65535	4981	39.03
65536	3743	36.76
65537	4981	39.03

We have used VHDL to design the presented multipliers by encoding an SD digit into a 2-bit binary code, and a simulation is performed under the condition of 1 μ CMOS gate array technology. Table 1 shows the performance of the modulo m multipliers used in an RNS having the moduli set of $\{2^{16}-1, 2^{16}, 2^{16}+1\} = \{65535, 65536, 65537\}$ the wordlength of which is equivalent to about 48-bit length in a binary number system.

The delay time of a 16-bit binary modulo m multiplier based on the architecture proposed in [4] is about 90ns under the same simulation condition, meaning that the presented modulo m SD multiplier with a binary tree of the modulo m SD adders is operating at very high speed.

4. ERROR DETECTION USING SD MODULAR ARITHMETIC CIRCUITS

The presented modulo m adder and multiplier can be applied to detect the error for the following product sum calculation,

$$Z = A \times B + C. \quad (14)$$

In the above equation, A and B are in the *n*-bit, C and Z are in the 2*n*-bit binary number representations, respectively. We have the following relationship with residue operations between the operands and the calculation result.

$$(Z-)_m \neq ((A)_m \times (B)_m + (C)_m)_m \quad (15)$$

where $m = 2^p + \mu$ and $\mu \in \{-1,1\}$ .. Z_m, A_m, B_m, C_m are in the p-digit SD number representation, respectively. When an error of the product-sum calculation results in $Z \neq A \times B + C$, if $(Z-)_m \neq ((A)_m \times (B)_m + (C)_m)_m$, then the error is detected.

Based on the above discussion, a product-sum circuit with an error checker is constructed as shown in Fig.5. There are a residue product-sum circuit and four binary-to-residue converters in the error checker. the binary-to-residue converter performs the conversion operation from A, for example, into $(A)_m$. When $E \neq 0$, the calculation error, which may occur in the product-sum circuit or in the error checker, is detected.

The following property is very useful for the binary-to-residue conversion.

Property 2: Let k be a positive integer. Then

$$(2^k)_m=(-\mu)^{(k \text{ div } p)}\times 2^{(k \bmod p)} \quad (16)$$

where $m = 2^p+\mu$, $\mu \in \{-1,1\}$and (k div p) is the integer part of the division result.

By the above property, for example, when n =16, p = 4 and $\mu = 1$,

$$\begin{aligned}(\sum_{i=0}^{15} a_i 2^i)_{2^{-}+1} &= ((-1)^{(12 \text{ div } 4)} \times \sum_{i=0}^{3} a_{i+12} 2^i \\ &+(-1)^{(8 \text{ div } 4)} \times \sum_{i=0}^{3} a_{i+8} 2^i \\ &+(-1)^{(4 \text{ div } 4)} \times \sum_{i=0}^{3} a_{i+4} 2^i \\ &+ \sum_{i=0}^{15} a_i 2^i)_{2^{-}+1} \\ &=(-(\sum_{i=0}^{3} a_{i+12} 2^i\end{aligned}$$

Therefore, a binary-to-residue converter can be constructed with an adder tree as shown in Fig.6. MUL is a p-by-1 digit multiplier.

Table 2 shows the performance of the residue arithmetic circuits designed in the residue checker with a modulus $m = 2^8+1$. The lengths of operands of the product-sum are 32-bit and 64-bit long, and the residue checker has an 8-digit SD representation. The Prod-sum is detected by the residue checker. The presented residue checker consists of one MSDA, one MSDM, two Converter (64) for 64-bit binary number to 8-digit residue SD number and two Converters (32) for 32-bit binary number to 8-digit residue SD number.

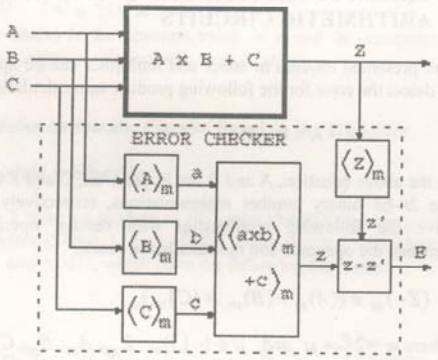


Figure 5: A product-sum circuit with a separate residue check.

Table 2 Performance of the checker

Circuit	Gates	DelayTime(ns)
Prod-sum	14.629	293.68
MSDA	190	5.45
MSDM	1552	18064
Convert(64)	1453	25.35
Convert(32)	669	16.9.3

Both the residue multiplier and the converters have the binary adder tree structure. Thus the presented residue arithmetic circuits for the error checker are much faster than that designed by the ordinary binary number arithmetic methods. For example, the delay timeof an 8-bit binary modulo m multiplier based on the architecture proposed in [5] is about 50ns under the same simulation condition, meaning that the presented residue SD arithmetic circuits with a binary tree of the modulo m SD adders are operating at very high speed.

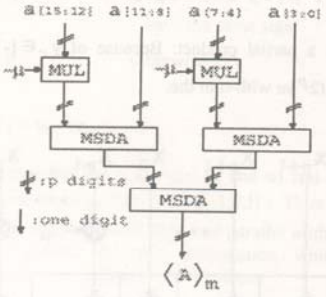


Figure 6: Binary-to-residue converter.

From Table 2, it is known that the hardware of the residue checker is about 40 % of the hardware the product-sum circuit has. It is preferable to use a small p to design a small error detection circuit, especially when we use multiple checkers with different moduli. However, considering the trade-off between the delay time by the stages of the binary adder tree and the detectablily of the residue converter, p is selected as a number proportional to n.

5. CONCLUSION

A fast modulo m ($m = 2^p \pm 1$) multiplier and a binary-to-residue converter have been proposed. Since they have a binary tree structure of radix-2 signed-digit number modulo m adders, the modulo m multiplication is performed in a time proportional to $\log_2 p$ and an n-bit binary number is converted into a p-digit SD residue number in a time proportional to $\log_2(n/p)$. The application to the error detection of a product-sum computation is considered by using the SD modulo m arithmetic circuits, and the presented error checker with the modular arithmetic circuits is faster than that with the binary modular arithmetic. However, the error checker may be costly in hardware. Our studies also focus on the error correction and the reduction of hardware.

REFERENCES

[1] N.S.Szabo and R.I.Tanaka , " Residue Arithmetic and Its

Applications to Computer Technology", New York : McGraw-Hill, 1967.

[2] M. A. Sonderstrand, W. K. Jendins, G. A. Junllien, and F. J. Taylor,"Residue Number System Arithmetic: Modern Applications in Digital Signal Processing," IEEE Press, New York, 1986.

[3] D.P. Agrawal and T.R.N.Rao,"Modulo (2^n+1) arithmetic logic,"IEE J. Electronic Circuits and Systems, vol.2, pp. 186-188, Nov. 1978.

[4] F.J.Taylor,"A VLSI residue arithmetic multiplier," IEEE Trans. Comput., vol.C-31, pp.540-546,June 1982.

[5] A.Hiasat,"New memoryless, mod $(2^n \pm 1)$ residue multiplier," Electron. Lett., vol.28, no.3, pp.314-315,Jan. 1992.

[6] A. Anizienis,"Arithmetic algorithms for errorcoded operands," IEEE Trans. Comput., vol.C22, no. pp.567-572, June 1973.

[7] T.R.N.Rao,"Biresidue error-correcting codes for computer arithmetic,", IEEE Trans. Comput., vol. C-19, no. pp.398-402, May 1970.

[8] A.Avizienis,"Signed-digit number representations for fast parallel arithmetic,"IRE Trans. Elect. Comput., EC-10,pp.389-400, Sept. 1961.

[9] S. Wei and K.shimizu,"A Novel Residue Arithmetic Hardware Algorithm Using a Signed-Digit Number Representation," IEICE Trans. Inf. & Syst. vol.E-83D, no.12, pp.2056-2064, Dec. 2000.

Convergence of Parallel Chaotic Generalized AOR Method for H-matrix

Dongjin Yuan
Department of Mathematics
Yangzhou University, Yangzhou, JiangSu, China
E-mail address: dongjinyuan@yahoo.com

ABSTRACT

In this paper we establish several algorithms of parallel chaotic generalized AOR iterative methods for solving large nonsingular systems based on some given models. Under some different assumptions of coefficient matrix A and its multisplittings we obtain corresponding sufficient conditions of convergence for some relaxed parameters.

Keywords: Multisplitting , parallel, chaotic, convergence, generalized AOR method, H-matrix.

1. INTRODUCTION

Parallel multisplitting iterative method for solving a large system of linear equations

$$Ax=b \tag{1}$$

Where $A\in R^{n\times n}$, $x\in R^n, b\in R^n$, take two basic forms, *synchronous* when all of the processor wait until they are updated with the results of the current iteration before they begin the next iteration or *asynchronous* when they act more or less independently of each other, using possibly delayed iterative values of the output of the other processors in computing their next iterate. In view of the potential time saving inherent in them, asynchronous iterative methods, or chaotic as they are often called, have attracted much attention since the early paper of Chazan and Miranker [2] introduced them in the context of point iterative schemes. Naturally, their convergence is of crucial interest and a number of convergence results (such as [1,3,4,5,6,13] etc.) have been obtained. In particular, the convergence of three relaxed chaotic parallel AOR methods have been investigated in references [6,13]. In this paper, we will establish some Algorithms of chaotic parallel-generalized AOR method and investigate their convergence for *H*-matrices.

Let us consider the splitting of the matrix A of (1) as follows:

$$A=D-C_L-C_U \tag{2}$$

with D=diag A and C_L, C_U strictly lower and upper triangular matrices obtained from A. The generalized accelerated overrelaxation (AOR) method, as in [12], is given by

$$x^{(k+1)}=F(\alpha,\Omega)x^{(k)}+(I-\alpha\Omega L)^{-1}b,$$
$$k=1,2,\cdots \tag{3}$$

where

$$F(\alpha,\ \Omega)=(I-\alpha\Omega L)^{-1}[I-\Omega+(1-\alpha)\Omega L+\Omega U]$$

is an iterative matrix and $L=D^{-1}C_L, U=D^{-1}C_U$, $\Omega=diag(\omega_1,\omega_2,\ldots,\omega_n)$ with $\omega_i\in R^+$ and α real parameter.

The matrix

B=L+U

is the Jacobi iterative matrix.

For $\alpha=\gamma/\omega$ and $\Omega=\omega I$, the generalized AOR method reduces to the AOR method with the iteration matrix $F(\gamma,\omega)=(I-\gamma L)^{-1}[(1-\omega)I+(\omega-\gamma)L+\omega U]$. The main purpose of this paper is to present several Algorithms of relaxed parallel chaotic generalized AOR schemes for solving large nonsingular system (1), in which the coefficient matrix A is an H-matrix, and investigate the corresponding convergence of these Algorithms.

2. NOTATION AND ALGORITHMS

Let us first introduce some of the notation and terminology, which will be used in this paper. For $x=(x_1,x_2,\cdots,x_n)^T\in R^n$ and

$A=(a_{ij})\in R^{n\times n}$ by $x\geq 0$ we mean that $x_i\geq 0$ for $i=1,\cdots,n$, and by $A\geq 0$ that $a_{ij}\geq 0$ for $i,j=1,\cdots,n$, in which case we say that x and A are nonnegative. For $A,B\in R^{n\times n}$, we write $A\leq B$ if $a_{ij}\leq b_{ij}$ hold for all entries of $A=(a_{ij})$ and $B=(b_{ij})$. By $|A|=(|a_{ij}|)$ we define the absolute value of $A\in R^{n\times n}$, it is a nonnegative matrix satisfying $|AB|\leq|A|\cdot|B|$. The notation $<A>=(<a>_{ij})$ represents the comparison matrix of $A\in R^{n\times n}$ where

$$<a>_{ij}=\begin{cases}|a_{ij}| & \text{if } i=j,\\ -|a_{ij}| & \text{if } i\neq j.\end{cases}$$

A matrix $A=(a_{ij})\in R^{n\times n}$ is an M-matrix if it is nonsingular with $A^{-1}\geq 0$ and $a_{ij}\leq 0$ for all $i\neq j$. It is an H-matrix if $<A>$ is an M-matrix, and an L-matrix if $a_{ii}>0$ for $i=1,\cdots,n$, and $a_{ij}\leq 0$ for all $i\neq j$.

A splitting of a matrix $A=(a_{ij})\in R^{n\times n}$ is a pair of matrices $M,N\in R^{n\times n}$ with $\det(M)\neq 0$ such that A=M-N. It is called a nonnegative splitting if $M^{-1}N\geq 0$ and an M-splitting if M is an M-matrix and $N\geq 0$. For any $s\geq 2$ a multisplitting of $A\in R^{n\times n}$ is a collection of s

triples (M_l, N_l, E_l) of $n \times n$ real matrices, l=1,2,$\cdots$,s, for which each E_l is nonnegative diagonal, each M_l is invertible and the equations

$$A = M_l - N_l, l = 1,2,\cdots,s \tag{4}$$

and

$$\sum_{l=1}^{s} E_l = I \tag{5}$$

are satisfied.

Using the given models in [1,6,13] and (3) we can now describe three Algorithms of relaxed parallel chaotic generalized AOR method by above notation.

Algorithm 2.1 Choose $x^{(0)} \in R^n$ *arbitrarily. For* $k=1,2,\cdots$, *until convergence, perform*

$$x^{(k+1)} = \sum_{l=1}^{s} E_l F_l^{\mu_l,k}(\alpha,\Omega,x^{(k)})$$

$$F_l(\alpha,\Omega,x^{(k)}) = (I-\alpha\Omega L_l)^{-1}[I-\Omega+(1-\alpha)\Omega L_l + \Omega U_l]x^{(k)} + (I-\alpha\Omega L_l)^{-1}b$$

with $\alpha \geq 0, \omega_i > 0, \mu_{l,k} \geq 1$.

where $F_l^{\mu_l,k}$ is the $\mu_{l,k}$ th composition of the affined mapping satisfying

$$F_l^{\mu_l,k} = \begin{cases} F_l \cdot F_l \cdots\cdots F_l & \mu_{l,k} \geq 1, \\ I & \mu_{l,k} = 0. \end{cases}$$

and

$$B = L_l + U_l, l = 1,2,\cdots,s.$$

By using a suitable positive relaxation parameter β, we then get the following relaxed Algorithm, which is based on Algorithm 2.1.

Algorithm 2.2 *Choose* $x^{(0)} \in R^n$ *arbitrarily. For* $k=1,2,\cdots$, *until convergence, perform*

$$x^{(k+1)} = \beta \sum_{l=1}^{s} E_l E_l^{\mu_l,K}(\alpha,\Omega,x^{(k)}) + (1-\beta)x^{(k)}$$

$$F_l(\alpha,\Omega,x^{(k)}) = (I-\alpha\Omega L_l)^{-1}[I-\Omega+(1-\alpha)\Omega L_l + \Omega U_l]x^{(k)} + (I-\alpha\Omega L_l)^{-1}b$$

with $\beta > 0, \alpha \geq 0, \omega_i > 0, \mu_{l,k} \geq 1$.

Next if we consider the case of relaxed chaotic generalized AOR method and assume that the index sequence $\{P_l\}$ is admissible and regulated, then we can get the following Algorithm.

Algorithm 2.3 *Choose* $x^{(0)} \in R^n$ *arbitrarily. For* $k=1,2,\cdots$, *until convergence, perform*

$$x^{(k+1)} = (I-\beta\sum_{l\in P_l})x^{(k)} + \beta\sum_{l\in P_l} E_l F_l^{\mu_l,k}(\alpha,\Omega,x^{(k-r_k+1)}).$$

$$F_l(\alpha,\Omega,x^{(k-r_k+1)}) = (I-\alpha\Omega L_l)^{-1}[I-\Omega + (1-\alpha)\Omega L_l + \Omega U_l]x^{(k-r_k+1)} + (I-\alpha\Omega L_l)^{-1}b$$

$$x^{(k-r_k+1)} = (x_1^{(k-r(1,k))}, x_2^{(k-r(2,k))}, \cdots, x_n^{(k-r(n,k))})^{\mathrm{T}}$$

with $\beta > 0, \alpha \geq 0, \omega_i > 0, \mu_{l,k} \geq 1$,

$\phi \neq P_l \subseteq \{1,\cdots,s\}$.

Now we can point that because the AOR method is only the special case of the generalized AOR method, the corresponding Algorithms in [6] and [13] are also the special cases of the above Algorithms 2.1-2.3.

3. CONVERGENCE OF THE ALGORITHMS

Before starting our convergence results concerning above Algorithms we should first introduce the following two lemmas, which have been presented in [13].

Lemma 3.1 *If A is an H-matrix, then*

(*a*) $|A^{-1}| \leq < A >^{-1}$;

(b) there exists a diagonal matrix P whose diagonal entries are positive such that AP is by rows strictly diagonally dominant, i.e.,

$$< A > Pe > 0 \tag{6}$$

with $e = (1,\cdots,1)^{\mathrm{T}}$.

Lemma 3.2 *Let A be an M-matrix, and let the splitting A=M-N be an M-splitting. If P is the diagonal matrix defined in Lemma 3.1, then*

$$\| P^{-1}M^{-1}NP \|_{\infty} < 1 \tag{7}$$

Theorem 3.3 *Let* $A \in R^{n\times n}$ *be an H-matrix and* $(D-(C_L)_l, (C_U)_l, E_l), l = 1,2,\cdots,s$, *be a multisplitting of A. Assume that for* $l = 1,2,\cdots,s$, *we have*

(1) $(C_L)_l$ *is the strictly lower triangular matrices and* $(C_U)_l$ *is the matrices such that the equalities* $A=D-(C_L)_l-(C_U)_l$ *hold.*

(2) $< A > = |D| - |(C_L)_l| - |(C_U)_l|$

$= |D| - |B_1|$, *where* $|B_1| = |(C_L)_l| + |(C_U)_l|$.

Then the sequence $\{x^{(k)}\}$ *generated by Algorithm 2.1 converges to the solution vector of system (1) for any starting vector* $x^{(0)} \in R^n$ *if* $(\alpha,\omega_i), i = 1,2,\cdots,$ $n, \in S_1$, *where*

$$S_1 = \{(\alpha,\omega_i) \in R^2 : 0 \leq \alpha \leq 1; \; 0 < \omega_i < 2/(1+\rho), i = 1,2,\cdots,n.\}$$

with $\rho = \rho(|D|^{-1}|B_1|)$.

Proof: Let us first denote $L_l = D^{-1}(C_L)_l$, $U_l = D^{-1}(C_U)_l$, $l = 1,2,\cdots,s$, then we can define the iterative matrix in the Algorithm 2.1

$$H(\alpha,\Omega)_k = \sum_{l=1}^{s} E_l\{(I-\alpha\Omega L_l)^{-1}[I-\Omega + (1-\alpha)\Omega L_l + \Omega U_l]\}^{\mu_l,k}$$

It is clearly that we need to find a constant σ with $0 \leq \sigma < 1$ and some norm, which are independent of k, such that for $k \geq 1, \| H(\alpha,\Omega)_k \| \leq \sigma$.

Since A is an H-matrix and for $l = 1,2,\cdots,s, L_l$ is a strictly lower triangular matrix, we see that each $< I-\alpha\Omega L_l >$ is

an M-matrix for $l = 1,2,\cdots,s$, and

$< I - \alpha\Omega L_l >^{-1} = (I - \alpha\Omega \mid L_l \mid)^{-1} \geq 0$.

Hence each $(I - \alpha\Omega L_l)$ is an H-matrix for $l = 1,2,\cdots,s$, and we have the following inequality

$\mid (I - \alpha\Omega L_l)^{-1} \mid \leq < I - \alpha\Omega L_l >^{-1} = (I - \alpha\Omega \mid L_l \mid)^{-1}$ From this relation it follows that

$$\begin{aligned} &\mid (I - \alpha\Omega L_l)^{-1}[I - \Omega + (1-\alpha)\Omega L + \Omega U] \mid \\ &\leq < I - \alpha\Omega L_l >^{-1} \mid I - \Omega + (1-\alpha)\Omega L_l + \Omega U_l \mid \\ &\leq (I - \alpha\Omega \mid L_l \mid)^{-1}[\mid I - \Omega \mid + \\ &\mid 1 - \alpha \mid \Omega \mid L_l \mid + \Omega \mid U_l \mid] \end{aligned}$$

Case: 1: $\omega_i \leq 1$. In this case , we denote

$$M_l(\alpha,\Omega) = I - \alpha\Omega \mid L_l \mid$$

and

$$N_l(\alpha,\Omega) = \mid I - \Omega \mid + (1-\alpha)\Omega \mid L_l \mid + \Omega \mid U_l \mid$$

Evidently, for $l = 1,2,\cdots,s$, we have the following relation

$$\begin{aligned} &M_l(\alpha, \Omega) - N_l(\alpha,\Omega) = \Omega - \Omega \mid L_l \mid - \Omega \mid U_l \mid \\ &= \Omega(I - \mid B \mid). \end{aligned}$$

Since, for $l = 1,2,\cdots,s, M_l(\alpha,\Omega)$ are M-matrices and

$N_l(\alpha,\Omega) \geq 0$, the splittings $M_l(\alpha,\Omega) - N_l(\alpha,\Omega)$ are M-splittings of the matrix $\Omega(I - \mid B \mid)$, which is M-matrix.

Case 2: $\omega_i > 1$. Suppose $\omega = \max_{1\leq i\leq n} \omega_i$.

We have

$$\begin{aligned} &\mid I - \Omega \mid + \mid 1 - \alpha \mid \Omega \mid L_l \mid + \Omega \mid U_l \mid \leq (\omega - 1)I + \\ &(1 - \alpha)\ \omega \mid L_l \mid + \omega \mid U_l \mid. \end{aligned}$$

and

$$(I - \alpha\Omega \mid L_l \mid)^{-1} \leq (I - \alpha\omega \mid L_l \mid)^{-1}$$

We also denote

$$M_l(\alpha,\omega) = I - \alpha\omega \mid L_l \mid$$

and

$$N_l(\alpha,\omega) = (\omega - 1)I + (1-\alpha)\omega \mid L_l \mid + \omega \mid U_l \mid$$ t

hen

$$M_l(\alpha,\omega) - N_l(\alpha,\omega) = (1 - \mid 1 - \omega \mid)\ I - \omega \mid B \mid$$

It is easy to prove (see from [6,13]) that

$(1 - \mid 1 - \omega \mid)I - \omega \mid B \mid$ is an M-matrix. Since, for $l = 1,2,\cdots,s$, $M_l(\alpha,\Omega)$ are M-matrices and $N_l(\alpha,\Omega) \geq 0$, the splittings $M_l(\alpha,\Omega) - N_l(\alpha,\Omega)$ are M-splittings of the matrix

$$(1 - \mid 1 - \omega \mid)I - \omega \mid B \mid.$$

Thus, for case 1 and 2, from Lemma 3.2 in the above it derives

$\| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty < 1, l = 1,2,\cdots,s.$

and hence

$$\begin{aligned} &P^{-1} \mid H(\alpha,\Omega)_k \mid Pe \\ &\leq \sum_{l=1}^{s} E_l \{P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P\}^{\mu_l,k} e \\ &\leq \sum_{l=1}^{s} E_l \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty^{\mu_l,k} e \\ &\leq \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty e. \end{aligned}$$

which implies

$$\begin{aligned} &\| P^{-1} H(\alpha,\Omega)_k P \|_\infty \\ &\leq \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty < 1. \end{aligned}$$

Consequently

$$\begin{aligned} &\mid H(\alpha,\Omega)_k \mid Pe = P(P^{-1} \mid H(\alpha.,\Omega)_k \mid P)e \\ &\leq P \| P^{-1} H(\alpha,\Omega)_k P \|_\infty e \\ &\leq \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty Pe, \\ &\quad l = 1,2,\cdots,s \end{aligned}$$

Let us denote

$\sigma = \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty$, then

$\| H(\alpha,\Omega)_k \| \leq \sigma < 1.$

We have completed the proof. □

Theorem 3.4 *Let $A \in R^{n\times n}$ be an H-matrix and $(D - (C_L)_l, (C_U)_l, E_l)$. $l = 1,2,\cdots,s$, be a multisplitting of A. Assume that for $l = 1,2,\cdots,s$, we have*

(1) $(C_L)_l$ is the strictly lower triangular matrices and $(C_U)_l$ is the matrices such that the equalities $A = D - (C_L)_l - (C_U)_l$ hold.

(2) $< A > = \mid D \mid - \mid (C_L)_l \mid - \mid (C_U)_l \mid = \mid D \mid - \mid B_1 \mid$, where $\mid B_1 \mid = \mid (C_L)_l \mid + \mid (C_U)_l \mid$.

(3) P is the diagonal matrix defined in Lemma 3.1 and $M_l(\alpha,\Omega), N_l(\alpha,\Omega)$ in Theorem 3.3.

Then the sequence $\{x^{(k)}\}$ generated by Algorithm 2.2 converges to the solution vector of system (1) for any starting vector $x^{(0)} \in R^n$ if

$(\alpha,\beta,\omega_i), i = 1,2,\cdots,n, \in S_2$,

where $S_2 = \{(\alpha,\beta,\omega_i) \in R^3 : 0 \leq \alpha \leq 1; 0 < \beta < 2$

$/(1+\theta); 0 < \omega_i < 2/(1+\rho), i = 1,2,\cdots,n.\}$

with $\rho = \rho(\mid D \mid^{-1} \mid B_1 \mid)$,

$\theta = \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega) N_l(\alpha,\Omega) P \|_\infty$

Proof Let us define the iterative matrix in the Algorithm 2.2

$$H(\alpha,\beta,\Omega)_k = \beta H(\alpha,\Omega)_k + (1-\beta)I$$

where

$$H(\alpha,\Omega)_k = \sum_{l=1}^{s} E_l \{(I - \alpha\Omega L_l)^{-1}[I - \Omega + (1-\alpha)\Omega L_l + \Omega U_l]\}^{\mu_l,k}$$

Similar to the proof of Theorem 3.3 we only need to prove that there exists a constant σ with $0 \leq \sigma < 1$, which is independent of *k*, such that

$$\| P^{-1} H(\alpha,\beta,\Omega)_k P \|_\infty \leq \sigma,$$

From the relation in the proof of Theorem 3.3,

$$\begin{aligned} &\| P^{-1} H(\alpha,\Omega)_k P \|_\infty \\ &\leq \max_{1\leq l\leq s} \| P^{-1} M_l^{-1}(\alpha,\Omega)_k N_l(\alpha,\Omega) P \|_\infty < 1, \end{aligned}$$

we obtain

$$\begin{aligned} &\| P^{-1} H(\alpha,\beta,\Omega)_k P \|_\infty \leq \beta \mid P^{-1} H(\alpha,\Omega)_k P \|_\infty \\ &\quad + \mid 1 - \beta \mid \end{aligned}$$

$\leq \beta \max_{1\leq i\leq s} \| P^{-1}M_i^{-1}(\alpha,\Omega)N_i(\alpha,\Omega)P \|_\infty$
$+|1-\beta|.$

Clearly, if $\omega_i (i=1,2,\cdots,n)$ and β satisfy the condition of this Theorem then

$\sigma \equiv \beta \max_{1\leq i\leq s} \| P^{-1}M_i^{-1}(\alpha,\Omega)N_i(\alpha,\Omega)P \|_\infty +$
$|1-\beta|<1.$

which completes the proof.

Using the proving process of Theorem 3.3, 3.4 and [13, Theorem 2.8] we get the following convergence of Algorithm 2.3.

Theorem 3.5 *Let* $A \in R^{n\times n}$ *be an H-matrix and* $(D-(C_L)_l, (C_U)_l, E_l), l=1,2,\cdots,s,$ *be a multisplitting of A. Assume that for* $l=1,2,\cdots,s,$ *we have*

(1) $(C_L)_l$ *is the strictly lower triangular matrices and* $(C_U)_l$ *is the matrices such that the equalities* $A=D-(C_L)_l-(C_U)_l$ *hold.*

(2) $<A>=|D|-|(C_L)_l|-|(C_U)_l|$
$=|D|-|B_l|$
where $|B_l|=|(C_L)_l|+|(C_U)_l|$.

(3) *P is the diagonal matrix defined in Lemma 3.1 and* $M_i(\alpha,\Omega), N_i(\alpha,\Omega)$ *in Theorem 3.3.*

(4) *The index sequence* $\{P_i\}$ *is admissible and regulated.*

Then the sequence $\{x^{(k)}\}$ *generated by Algorithm 2.3 converges to the solution vector of system (1) for any starting vector* $x^{(0)} \in R^n$ *if* $(\alpha, \beta, \omega_i)$, $i=1,2,\cdots,n, \in S_3$,

where

$S_3 = \{(\alpha,\beta,\omega) \in R^3 : 0\leq\alpha\leq 1; 0<\beta<2/(1+\theta); 0<\omega_i<2/(1+\rho), i=1,2,\cdots,n.\}$ with

$\rho = \rho(|D|^{-1}|B_l|),$
$\theta = \max_{1\leq i\leq s} \| P^{-1}M_i^{-1}(\alpha,\Omega)N_i(\alpha,\Omega)P \|_\infty .$

This completes the Theorem.

Since the size of paper is limited, we will discuss the practical use of the Algorithms in detail in another paper.

REFERENCES

[1] R. Bru, L. Elsner and M. Neumman, *Models of parallel chaotic iterative methods*, Linear Algebra Appl., 103(1988), 175-192

[2] D. Chazan and W. Miranker, *Chaotic relaxation*, Linear Algebra Appl., 2(1969), 199-222.

[3] L. Elsner, M. Neumman and B. Vemmer, *The effect the number of processors on the convergence of the parallel block Jacobi method*, Linear Algebra Appl., 154/156(1991), 311-330.

[4] L. Elsner and M. Neumman, *Monotonic sequences and rates of convergence of asynchronized iterative methods*, Linear Algebra Appl., 180(1993), 17-33.

[5] A. Frommon and G. Mayer, *Convergence of relaxed parallel multisplittings methods*, Linear Algebra Apple, 119 (1989), 141-152.

[6] P.E. Kloeden and D. Yuan, *Convergence of relaxed chaotic parallel iterative methods*, Bulletin of Austral. Math. , Soc., 50(1994), 167-176.

[7] L. Li, *Convergence of asynchronous iteration with arbitrary splitting form*, Linear Algebra Appl., 113(1989), 119-127.

[8] M. Neumman and R.J. Plemmons, *Convergence of parallel multisplitting iterative methods for M-matrices*, Linear Algebra Appl., 88/89(1987), 559-573.

[9] D. P. O'Leary and R. E. White, *Multisplittings of matrices and parallel solution of linear systems*, SIAM.J.Algebraic Discrete Methods., 6(1985), 630-640.

[10] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, San Francisco, London, 1970.

[11] Y.Song, *Convergence of parallel multisplitting method for H-matrices*, Intern. J.Computer Math., 48(1993).

[12] Y.Song, *Convergence of the generalized AOR method*, Linear Algebra Appl., 256 (1997), 199-218.

[13] Y. Song and D. Yuan, *On the convergence of relaxed parallel chaotic iterations for H-matrix*, Intern.J.Computer Math., 52(1994), 195-209.

[14] R.Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, N.J. 1962.

[15] D. Wang, *On the convergence of the parallel multisplitting AOR algorithm*, Linear Algebra Appl., 154/156(1991), 473-486.

[16] D. M.Young, Iterative Solution of Large Linear Systems, Academic, 1971.

A Preconditioner Derived from Finite Difference Equations for Solving Poisson's Equations*

Yaming Bo, Anderas Lauer, Anderas Wien and Peter Waldow
Institute of Mobile and Satellite Communication Techniques
47475 Kamp-Lintfort, Germany

ABSTRACT

Based on the finite difference equations of the Poisson's boundary problem, the matrix of the linear system is expressed with the so-called difference operator matrices, which are rectangular, and the spacing matrices. Then, the equivalent expression with the summation of the products of square matrices is derived. And a sparse preconditioner is constructed with the square matrices, which is superior in acceleration effect to the preconditioners of incomplete LU (ILU) decomposition and modified ILU (MILU) decomposition. The parallelization of the algorithm is possible.

Keywords: Iterative Method, Preconditioner, Conjugate Gradient Method, Elliptic Differential Equation. Solver Implementation.

1. INTRODUCTION

In order to establish a multi-layer structure simulation library, which will be employed to design integrated passive components, a solver for the solution of the Poisson's equation is developed. In consideration of the large number of unknowns, iterative methods are adopted. The total memory space and time-consuming should be considered for the solver. Another aspect of the selected algorithm which should be considered is the non-uniform discretization. The successive overrelaxation (SOR) method, in which the optimal factor can be approached in iterations [1], is a proper algorithm choice because of the simple evaluation formula and less memory requirements. The lower memory expense can also accelerate the convergence to some extent because of the less data exchange between memory and cache. However, the slow convergence of this method still results in too much computational time for millions of unknowns.

To reduce the CPU time for the solution or to give an option of time priority in the solver, a new preconditioner for the conjugate gradient method (CGM) is presented in this paper, which is derived from the finite difference equations with non-uniform meshes and possesses a satisfactory acceleration effect as well as the low memory expense. The performances of the preconditioner are verified by the numerical results and compared with those of the incomplete LU (ILU) decomposition and the modified ILU (MILU) decomposition preconditioners [2-3]. The formulation may be regarded as a technique to construct preconditioners from the finite difference equations for the numerical solutions of the similar differential equations. The extension for internal boundaries, which is important for a solver for engineering problems, is considered. The parallelism of this algorithm could be at least the same as those of ILU and MILU preconditioners.

* This work is supported by Alfried Krupp von Bohlen und Halbach Foundation of Germany.

2. FORMULATION

Difference Operator Matrix and Spacing Matrix

Poisson's equation for circuit parameter extraction is solved in a rectangular or cuboidal region. Non-uniform meshes are used for the discretization. For simplicity, 1-dimensional Poisson's equation

$$\frac{d^2\varphi}{dx^2} = \rho \tag{1}$$

with Dirichlet boundary condition is considered at first. It is well known that the 3-point finite difference equations with non-uniform meshes could be expressed as

$$2\frac{h_i\varphi_{i-1} + h_{i-1}\varphi_{i+1} - (h_{i-1}+h_i)\varphi_i}{h_{i-1}h_i(h_{i-1}+h_i)} = \rho_i, \tag{2}$$

where $i = 1,2,\cdots n$. That means that the nodes are in natural ordering and there are n internal nodes between two end-points φ_0 and φ_{n+1}. Eq.(2) can be rewritten in the form

$$-\frac{\varphi_{i-1}}{h_{i-1}} + (\frac{1}{h_{i-1}} + \frac{1}{h_i})\varphi_i - \frac{\varphi_{i+1}}{h_i} = -\frac{(h_{i-1}+h_i)}{2}\rho_i \tag{3}$$

By moving the terms of φ_0 and φ_{n+1} to the right hand side of Eq.(3), one can get a matrix-vector equation as

$$\mathbf{Au} = \mathbf{b}. \tag{4}$$

The matrix $\mathbf{A}$ is positive definite and can be evaluated with the formula

$$\mathbf{A} = \mathbf{LDL}^T, \tag{5}$$

where the superscript T means transpose, the $(n+1)\times(n+1)$ matrix $\mathbf{D}$ is diagonal and called *spacing matrix*, of which the diagonal consists of the reciprocals of the mesh sizes h_i. The matrix

$$\mathbf{L} = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix}_{n\times(n+1)} \tag{6}$$

is an $n\times(n+1)$ matrix and called *difference operator matrix*. Eq.(5) can be directly derived from the left hand side of Eq.(3). And Eq.(4) together with Eq.(5) is just the matrix form of Eq.(3). Thus, the differential operator in Eq.(1) together with the boundary condition is transformed into a discrete form.

Equivalent Square Matrix

In the matrix form of Eq.(3), i.e. Eq.(4), the product of the matrices $\mathbf{L}$ and $\sqrt{\mathbf{D}}$ is not square, so it cannot be used as the matrix for a preconditioner directly. Now, let $\tilde{\mathbf{L}}$ be a $n\times n$ square triangular matrix in the form

$$\tilde{\mathbf{L}}=\begin{bmatrix} l_1 & 0 & 0 & \cdots & 0 & 0 \\ -\alpha_1 & l_2 & 0 & \cdots & 0 & 0 \\ 0 & -\alpha_2 & l_3 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & l_{n-1} & 0 \\ 0 & 0 & 0 & \cdots & -\alpha_{n-1} & l_n \end{bmatrix}_{n\times n} \tag{7}$$

with the relationship

$$\tilde{\mathbf{L}}\tilde{\mathbf{L}}^T=\mathbf{L}\mathbf{D}\mathbf{L}^T=\mathbf{A}, \tag{8}$$

the nonzero elements of the square matrix can be evaluated by using

$$\begin{cases} l_1=\sqrt{s_0+s_1} \\ \alpha_i=\dfrac{s_i}{l_i} \\ l_{i+1}=\sqrt{s_i+s_{i+1}-\alpha_i^2} \end{cases}, \quad i=1,2,\cdots,n-1 \tag{9}$$

where s_i are the reciprocals of the mesh steps h_i. Thus, a square matrix $\tilde{\mathbf{L}}$ which is equivalent to the product of $\mathbf{L}$ and $\sqrt{\mathbf{D}}$ is derived and can be used to express the equation matrix $\mathbf{A}$. By use of this square matrix, one can directly obtain the solution of the linear system (4) for 1-dimensional problem.

Two-dimensional Problems

The derivation above can be easily extended to the Poisson's boundary problems with more dimensions. The 5-point finite difference equations from 2-dimensional problem in a rectangular region can be written as

$$\begin{aligned} &c_{y,j}[-s_{x,i-1}\varphi_{i-1,j}+(s_{x,i-1}+s_{x,i})\varphi_{i,j} \\ &-s_{x,i}\varphi_{i+1,j}]+ \\ &c_{x,i}[-s_{y,j-1}\varphi_{i,j-1}+(s_{y,j-1}+s_{y,j})\varphi_{i,j} \\ &-s_{y,j}\varphi_{i,j+1}] \\ &=-\frac{c_{x,i}c_{y,j}}{2}\rho_{i,j} \end{aligned}, \tag{10}$$

where

$$h_{x,i}=\frac{1}{s_{x,i}}, \quad i=0,1,2,\cdots,n \tag{11}$$

and

$$h_{y,j}=\frac{1}{s_{y,j}}, \quad j=0,1,2,\cdots,m \tag{12}$$

are the mesh sizes along x and y directions, respectively, and

$$c_{x,i}=h_{x,i-1}+h_{x,i}, \quad i=1,2,\cdots,n, \tag{13}$$

$$c_{y,j}=h_{y,j-1}+h_{y,j}, \quad j=1,2,\cdots,m. \tag{14}$$

Then, the equation matrix $\mathbf{A}$ with respect to the Eq.(10) can be written as

$$\mathbf{A}=\mathbf{L}_X\mathbf{D}_X\mathbf{L}_X^T+\mathbf{L}_Y\mathbf{D}_Y\mathbf{L}_Y^T \tag{15}$$

if the nodes are in natural ordering, where

$$\mathbf{D}_X=diag(c_{y,1}\mathbf{D}_{sx},c_{y,2}\mathbf{D}_{sx},\cdots,c_{y,m}\mathbf{D}_{sx}), \tag{16}$$

$$\mathbf{D}_Y=diag(s_{y,0}\mathbf{D}_{cx},s_{y,1}\mathbf{D}_{cx},\cdots,s_{y,m}\mathbf{D}_{cx}), \tag{17}$$

$$\mathbf{D}_{sx}=diag(s_{x,0},s_{x,1},\cdots,s_{x,n}), \tag{18}$$

and

$$\mathbf{D}_{cx}=diag(c_{x,1},c_{x,2},\cdots,c_{x,n}). \tag{19}$$

The difference operator matrix along x direction $\mathbf{L}_X$ is a block diagonal matrix, of which all the diagonal blocks are the same and in the form of Eq.(6). Another difference operator matrix along y direction is

$$\mathbf{L}_Y=\begin{bmatrix} -\mathbf{I}_n & \mathbf{I}_n & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{I}_n & \mathbf{I}_n & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\mathbf{I}_n & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{I}_n & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I}_n & \mathbf{I}_n \end{bmatrix}_{mn\times n(m+1)} \tag{20}$$

where $\mathbf{I}_n$ is the $n\times n$ identity matrix.

By the similar derivation for 1-dimensional problem, we can let square lower triangular matrices

$$\tilde{\mathbf{L}}_X=diag(\sqrt{c_{y,1}}\mathbf{L}_x,\sqrt{c_{y,2}}\mathbf{L}_x,\cdots,\sqrt{c_{y,m}}\mathbf{L}_x) \tag{21}$$

and

$$\tilde{\mathbf{L}}_Y=\begin{bmatrix} l_1^y\mathbf{D}_{cx}^h & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ -\alpha_1^y\mathbf{D}_{cx}^h & l_2^y\mathbf{D}_{cx}^h & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & l_{m-1}^y\mathbf{D}_{cx}^h & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & -\alpha_{m-1}^y\mathbf{D}_{cx}^h & l_m^y\mathbf{D}_{cx}^h \end{bmatrix} \tag{22}$$

satisfy

$$\tilde{\mathbf{L}}_X\tilde{\mathbf{L}}_X^T=\mathbf{L}_X\mathbf{D}_X\mathbf{L}_X^T \tag{23}$$

and

$$\tilde{\mathbf{L}}_Y\tilde{\mathbf{L}}_Y^T=\mathbf{L}_Y\mathbf{D}_Y\mathbf{L}_Y^T \tag{24}$$

respectively,

where

$$\mathbf{L}_x=\begin{bmatrix} l_1^x & 0 & \cdots & 0 & 0 \\ -\alpha_1^x & l_2^x & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & l_{n-1}^x & 0 \\ 0 & 0 & \cdots & -\alpha_{n-1}^x & l_n^x \end{bmatrix}, \tag{25}$$

and

$$\mathbf{D}_{cx}^h=\sqrt{\mathbf{D}_{cx}}, \tag{26}$$

Then the recursive formulae similar to Eq.(9) for the elements of Eqs.(22) and (25) can be obtained as

$$\begin{cases} l_1^x=\sqrt{s_{x,0}+s_{x,1}} \\ \alpha_i^x=\dfrac{s_{x,i}}{l_i^x} \\ l_{i+1}^x=\sqrt{s_{x,i}+s_{x,i+1}-(\alpha_i^x)^2} \end{cases}, i=1,2,\cdots,n-1 \tag{27}$$

$$\begin{cases} l_1^y = \sqrt{s_{y,0} + s_{y,1}} \\ \alpha_j^y = \dfrac{s_{y,j}}{l_j^y} \qquad , j = 1,2,\cdots m-1 \\ l_{j+1}^y = \sqrt{s_{y,j} + s_{y,j+1} - (\alpha_j^y)^2} \end{cases} \tag{28}$$

Thus, the split parts of the matrix $\mathbf{A}$ at the right hand side of Eq.(15) can be replaced with the products of these matrices as

$$\mathbf{A} = \tilde{\mathbf{L}}_x \tilde{\mathbf{L}}_x^T + \tilde{\mathbf{L}}_y \tilde{\mathbf{L}}_y^T \tag{29}$$

Finally, we simply construct the preconditioner for the CGM as

$$\mathbf{P} = (\tilde{\mathbf{L}}_x + \tilde{\mathbf{L}}_y)^{-1} \tag{30}$$

It can be found out from Eqs.(21)-(22) and (29) that the elements of the matrix $\mathbf{A}$ and the matrix $\tilde{\mathbf{L}}_X + \tilde{\mathbf{L}}_Y$ for the preconditioner can be determined by only $3(n+m)-2$ values, which are $c_{x,j}$, $c_{y,j}$, l_i^x, α_i^x, l_j^y and α_j^y. Therefore, the number of the memory units for the equation matrix and the preconditioner is just about $3(n+m)$, which could be very smaller than $n \times m$, the number of the memory units of a vector, for large n and m. This fact is useful for the tradeoff between the memory space and the computing speed. The block forms of the matrices $\tilde{\mathbf{L}}_X$ and $\tilde{\mathbf{L}}_Y$ may be employed for parallelization of this preconditioned algorithm.

3. CONSIDERATIONS OF EXTENSION

The derivation of the preconditioner for 3-dimensional problem in a cuboidal region is just the same as that for 2-dimensional problem. The finite difference equation for 3-dimensional problem can be written as

$$\begin{aligned} & c_{y,j}c_{z,l}[-s_{x,i-1}\varphi_{i-1,j,l} + (s_{x,i-1} + s_{x,i})\varphi_{i,j,l} \\ & - s_{x,i}\varphi_{i+1,j,l}] + \\ & c_{x,i}c_{z,l}[-s_{y,j-1}\varphi_{i,j-1,l} + (s_{y,j-1} + s_{y,j})\varphi_{i,j,l} \\ & - s_{y,j}\varphi_{i,j+1,l}] + \\ & c_{x,i}c_{y,j}[-s_{z,l-1}\varphi_{i,j,l-1} + (s_{z,l-1} + s_{z,l})\varphi_{i,j,l} \\ & - s_{z,l}\varphi_{i,j,l+1}] \\ & = -\frac{c_{x,i}c_{y,j}c_{z,l}}{2}\rho_{i,j,l} \end{aligned} \tag{31}$$

Eq.(31) also implies the fact that the equation matrix is

$$\mathbf{A} = \mathbf{L}_x \mathbf{D}_x \mathbf{L}_x^T + \mathbf{L}_y \mathbf{D}_y \mathbf{L}_y^T + \mathbf{L}_z \mathbf{D}_z \mathbf{L}_z^T, \tag{32}$$

where the spacing matrices and the difference operator matrices should be redefined with $c_{x,i}$, $c_{y,j}$, $c_{z,l}$, $s_{x,i}$, $s_{y,j}$ and $s_{z,l}$. The detailed expressions of these matrices are omitted for conciseness. It should be noted that the difference operator matrices $\mathbf{L}$ in Eq.(32) are rectangular. The equivalent expressions with square matrices for three summation parts of matrix $\mathbf{A}$ can also be given in the same way for 2-dimensional problem. That means the equation matrix $\mathbf{A}$ for 3-dimensional problem can also be expressed with some square matrices as

$$\mathbf{A} = \tilde{\mathbf{L}}_x \tilde{\mathbf{L}}_x^T + \tilde{\mathbf{L}}_y \tilde{\mathbf{L}}_y^T + \tilde{\mathbf{L}}_z \tilde{\mathbf{L}}_z^T. \tag{33}$$

There exists the same relationship as Eqs.(27) and (28) between the elements of the square matrices in Eq.(33) and the spacing and the difference operator matrices in Eq.(32). Thus, the preconditioner can be chosen as

$$\mathbf{P} = (\tilde{\mathbf{L}}_x + \tilde{\mathbf{L}}_y + \tilde{\mathbf{L}}_z)^{-1}, \tag{34}$$

It should be emphasized that the basic idea for preconditioner construction is to find an equivalent expression with square matrices for matrix $\mathbf{A}$ based on its split form derived from the finite difference equations. This idea is universal for different numbers of dimensions.

In order to use this algorithm in a practical solver for multi-layer structures, the internal boundary is necessary to be considered. We would simply pointed out that, the split expression of Eq.(29) or Eq.(32) is also valid for a structure with internal boundaries in the rectangular or cuboidal region if all the boundaries are coincident with some grid lines. This can be guaranteed by dividing the internal boundaries with orthogonal polygonal lines to make an approximate discretization. The manipulation of the boundaries and the ordering of the nodes are mainly the works of programming for the solver implementation.

The recursive evaluation Eq.(9) could become a direct assignment for some continuous nodes which are spaced uniformly. This is useful for automatic discretization, with which the meshes could be partly uniform for most nodes. A large number of matrix elements, which are the same as one of them, could be unnecessary to be stored in computer memory. The direct form of Eq.(9) for uniformly spaced nodes can easily satisfy this requirement of implementation.

4. NUMERICAL RESULTS

The preconditioner presented has been tested for solver implementation and compared with incomplete LU (ILU) decomposition and modified ILU (MILU) decomposition preconditioners. Some of the test results are listed as follows. In order to compare the performances of the algorithms, ten examples with the random (non-uniform) mesh sizes are given in Table 1, and the corresponding iteration numbers for the solutions with the precision of 1.0×10^{-20} are shown in Table 2. It could be found from the data that the algorithm with the preconditioner $\mathbf{P}$ converges most rapidly among three kinds of iterative procedures.

Table 1. The random mesh sizes of the 2-dimensional test examples.

No.	$s_{x,i}, i = 0,1,2,\cdots,19$	$s_{y,j}, j = 0,1,2,\cdots,13$
1	{4,2,1,3,4,4,3,3,4,2,3,4,3,2,1,3,1,4,4,1}	{2,4,3,1,3,3,2,3,1,2,3,2,2,4}
2	{4,4,3,4,4,4,3,2,4,3,1,2,2,3,3,4,4,1,3,4}	{2,3,2,2,3,4,3,4,4,2,4,4,1,1}
3	{3,2,2,3,1,4,2,4,2,3,1,2,4,1,3,4,1,4,2,4}	{2,3,4,1,4,4,4,3,4,2,3,4,1,1}
4	{3,2,3,2,1,1,4,2,3,4,2,2,4,1,2,4,3,4,2,3}	{4,4,2,4,4,4,2,2,2,1,2,3,2,4}
5	{2,4,3,4,2,1,1,1,2,4,3,1,4,2,2,3,4,1,2,2}	{3,1,1,2,2,3,1,4,2,4,1,3,1,3}
6	{3,2,2,4,3,3,4,4,1,3,3,2,1,1,1,1,2,4,4,2}	{2,2,3,3,1,2,1,1,3,1,1,2,3,3}
7	{2,2,3,4,3,3,2,4,1,1,1,2,4,4,3,4,4,2,3,3}	{3,1,4,1,1,1,2,3,1,4,1,2,4,3}
8	{3,1,3,1,3,1,4,3,1,1,3,4,2,2,2,2,3,3,3,4}	{4,2,1,3,4,1,2,4,2,3,4,1,1,2}
9	{1,4,2,2,1,4,1,2,2,2,3,1,4,2,2,2,4,2,3,4}	{1,4,2,3,2,4,2,2,3,3,3,1,2,3}
10	{1,1,1,1,3,2,2,2,4,2,4,1,1,2,2,4,2,4,2,1}	{2,2,3,3,1,3,4,3,1,1,1,2,3,1}

Table 2. The iteration numbers for the solutions of the examples with a relative residual less than 1.0×10^{-20}

No.	CGM	Preconditioner		
		P	ILU	MILU
1	130	37	41	38
2	125	36	39	37
3	126	36	40	37
4	135	37	41	40
5	136	37	43	38
6	141	38	41	41
7	142	37	42	39
8	139	36	42	38
9	123	35	41	38
10	129	36	38	39

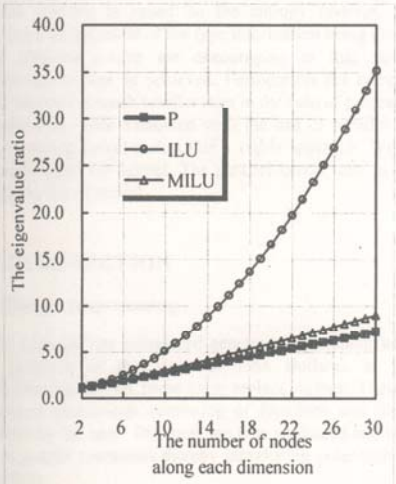


Fig.1 The eigenvalue ratio for 2-dimensional problems with uniform meshes

In order to observe the asymptotic properties of the preconditioned algorithm from the numerical experiments, 2-dimensional and 3-dimensional problems with uniform meshes and different node numbers are calculated. Fig.1 and Fig.2 show the ratios of maximal and minimal eigenvalues of the matrix product $\mathbf{PAP}^T$. The eigenvalue ratio is used as a measurement of the acceleration effect. It can be seen from the curves that the preconditioner presented in this paper possesses better acceleration effect than the other two for both 2-dimensional and 3-dimensional problems. The difference of the eigenvalue ratios between MILU preconditioner and **P** for 3-dimensional problem is more obvious than that for 2-dimensional problem.

The problems with uniform meshes are solved with the boundary conditions of unity at one side and zero at other sides. The precision for the termination of the iteration is also

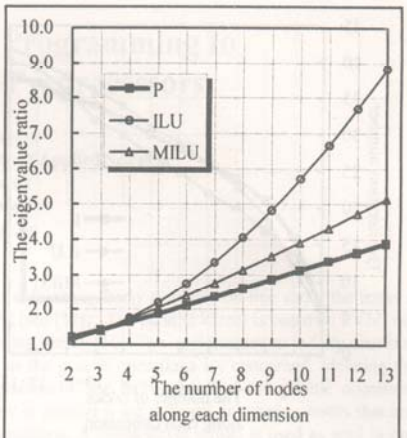


Fig.2 The eigenvalue ratio for 3-dimensional problems with uniform meshes

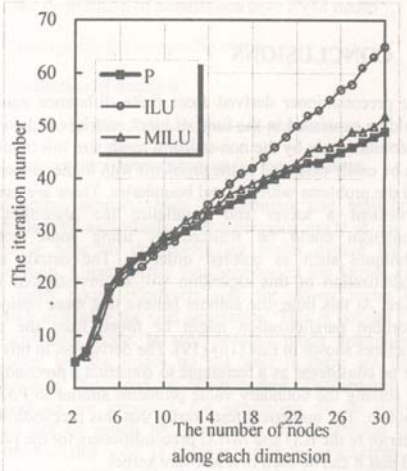


Fig.3 The iteration number for 2-dimensional problems with uniform meshes

1.0×10^{-20}. The results are shown in Fig.3 and Fig.4, from which one can find that the iteration number of the preconditioner **P** increases more slowly with the node number than those of ILU and MILU preconditioners. The tendencies of the curves tell the same fact as the eigenvalue ratios do. However, in some cases of small node number, the iteration number of the preconditioner **P** is greater than that of ILU preconditioner. It is due to the well-known fact that the convergence rate depends not only on the eigenvalue ratio but also on the distribution of all the eigenvalues. With the increase of the node number, the algorithm with preconditioner **P** will converges faster than others.

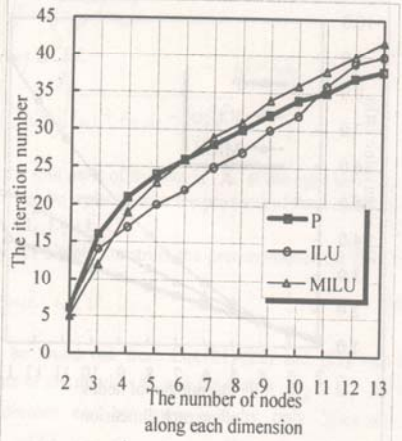


Fig.4 The iteration number for 3-dimensional problems with uniform meshes.

5. CONCLUSIONS

The preconditioner derived from finite difference equations could be expressed in the form of block matrices and could be determined only by the non-uniform mesh size information. It can be easily extended to the problems with higher dimensions and the problems with internal boundaries. These are useful to implement a solver and parallelize the algorithm. The parallelism could be realized by using some common techniques such as colored ordering. The details of the parallelization of this algorithm will be investigated in the future. At this time, the authors believe that more options for algorithm parallelization might be found from the matrix structures shown in Eqs.(16)-(19). The derivation in this paper can be considered as a technique to construct a preconditioner for solving the boundary value problems similar to Poisson's problem. The numerical tests verify that this preconditioner is superior to the ILU and MILU preconditioners for the problem and that it can be used in a software kernel.

REFERENCES

[1] D. M. Young and R. Y. Gregory, A Survey of Numerical Mathematics, in two volumes, Vol.II, Dover Pub., Inc., New York: Dover Publications, Inc., 1988.

[2] D. E. Keyes, A. Samed and V. Venkatakrishnan, Parallel Numerical Algorithms, ICASE/LaRC Interdisciplinary Series in Science and Engineering, Vol.4, Kluwer Academic, Dordecht, 1997.

[3] G. H. Golub and C. F. Van Loan, Matrix Computations (2nd edition), The Johns Hopkins University Press, Baltimore and London, 1989.

Using Parallel Genetic Programming to Evolve Compression Preprocessors

Johan J.M.G. Parent
Vrije Universiteit Brussel, Faculty of Applied Sciences
4K216, Pleinlaan 2
1150 Brussels, Belgium
parentjo@vub.ac.be

ABSTRACT

We present a new approach for applying genetic programming to lossless data compression. Unlike programmatic compression the evolved programs are preprocessors. The fitness function is based on the entropy concept, thereby making it independent of the type information being processed. The obtained results are encouraging in that significant improvements can be achieved. Furthermore the computation time required is much smaller than in the case of programmatic compression. This combined with the use of parallel genetic programming kernel makes this a viable approach. We used a strongly typed GP kernel. The parallel kernel runs in parallel using the island model.

1. INTRODUCTION

1.1 Genetic programming

John Koza [8] has introduced genetic programming which is an extension of the ideas of John Holland. In genetic programming the bit parse trees replace strings. These trees represent expressions consisting of functions and terminals defined by the user. The trees are then evaluatetl and subject-ed to genetic operations thereby introducing potentially better solutions.

1.2 Parallel genetic programming

Parallel genetic programming is not a new rescarch domain. Genetic algorithms and genetic programming are well suited for parallelization when using the so-called island model. In this model several populations (demes) evolve simultaneously and periodically exchange individuals. The easiest parallelization distributes the populations onto different computing nodes. Thereby allowing the evolution of the populations to proceed in parallel. The infrequent communication between populations and the usually computationally expensive evaluation of individuals makes parallel genetic programming an example of coarse grain parallelism.
The island model however introduces new parameters in the genetic programming scheme. These parameters are the exchange frequency, number of individual exchanged, number of populations and the exchange topologies. The implications of this model have been studied by many researcher: Punch[14], Koza and Andre [1], . . .

2. PARALLEL KERNEL

The package that has been used for this experiment is a modified version of the Lil-gp [13] that can run in parallel. The software is freely available on-line under the terms of the GPL (see [12]).. The parallel kernel is based on PVM message passing library [10]. The implementation is fully distributed as far as the genetic operations are concerned. Meaning that the evaluation of the individuals and the genetic operations are done in parallel n the different computing nodes that host the populations. A client-server model is used as well in order to collect statistics about the experiments and to configure the computing nodes. The parallel kernel transparently handles several tasks:

- the distribution of populations onto PVM nodes
- configuring the computing nodes
- collection of statistics
- exchange of individuals between populations
- peer-to-peer communication for the exchanges

The kernel supports both synchronous and asynchronous exchanges of individuals. In synchronous mode at any given point in time all the computing nodes evolve the same generation. Using barrier synchronization the kernel ensures that faster nodes effectively wait for slower nodes before starting to evolve the next generation. This is not the case in asynchronous mode. In asynchronous mode individuals are exported but the nodes do not wait if there are no new individuals that need to be imported. Furthermore when individuals are imported, the gencration of the individual needs not to match the generation being evolved at that moment on that node. In order to match the granularity of the parallel hardware the kernel is capable of grouping several populations on one computing node. In that case, he kernel distinguishes between internal and external interoperation exchanges to minimize the use of the message-passing library.
The main advantage of the parallel kernel is the transparent handling of all the parallel operations. Yet the computational speedups one achieves using the island model are significant. Thanks to the design of the kernel the user needs not to worry about this during the implementation of the problem!

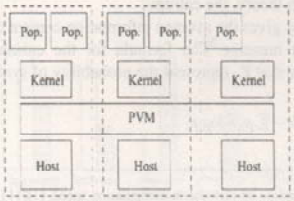


Figure 1: The parallel kernel completely shields the user from the parallel operations. Handling the population mapping, exchange of individuals and the statistics.

3. EA'S AND COMPRESSION

Evolutionary algorithms have been used in the past for compression purposes. Two approaches can be distinguished. In a first group, genetic algorithms were used to find parameters for a compression algorithm in order to maximize compression Driesen[4]. Feiel and Ramakrishnan [6] have used genetic algorithms to optimize the compression of color images using vector quantization. Other approaches used genetic programming for what is called programmatic compression. De Falco and al.[5] have used genetic programming for string compression. Fukunaga and Stechert[7] have used genetic programming for lossless compression of grayscale images. Nordin and Banzhaf[11] achieved lossy programmatic compression of images and sound. Noteworthy is the fact that [11] [7] both used a genocompiler for their experiments. This software eliminates the function call overhead incurred by other systems during the evaluation of the individuals. Luke[9] reports a relative improvement in speed up of 2000 times compared to LISP code and of 100 times compared to interpreted C (the latter has been used for this experiment).

3.1 Preprocessing

Our attention here goes toward lossless compression algorithms. Examples of popular lossless compression programs are gzip and Winzip [3]. One of the more recent contributions in generic lossless compression is the Burrows-Wheeler transform (1993) [2].[1]

Instead of aiming for a program that recodes the data, we seek for a transformation. We investigate whether a program can be evolved to transform given data in order to enhance its compressibility. The exact transformation is not explicitly known but one can formulate certain conditions it should possess. Such a preprocessing program could be formalized as a function P that works on a string S over an alphabet A . The length of a string S is denoted as | S|. C is represents a data compression algorithm.
The result we are looking for is a transformation P , such that the following condition holds:

$$S- = P(S) \tag{1}$$

$$|C(S-)| < |C(S)| \tag{2}$$

Stating the problem in these terms makes it easy to fold it into the genetic programming framework since both the program we are looking for, P here, and the fitness function are easily separated.
The conditions has two serious disadvantages though. First, computing the result of Eq. (2) is computationally expensive. Second, the transformation depends on the compression algorithm used.
To avoid this problem we reformulated it using a metric from information theory, the entropy.
The entropy gives the average information content of a symbol in a given message.[2]The formula for the entropy is given below, note that P_i represent the probability of symbol i in the message.[3]

$$H = -\sum p_i \bullet \log p_i \tag{3}$$

[1] Notwithstanding he increase in computation speed, [7][11] report runs lasting several tens of hours on powerful workstations..
[2] Entropyhas a much more rigorous mathematical foundation but the description given here suffices for the purpose of this text.
[3] The model used here is a _rst order model (marginal probability). Higher order models are based on conditional probabilities.

Using the entropy (Eq (3)) we have a means to determine how much information is actually present in a given message. One can use the entropy as an objective criterion for the transformation that we seek to evolve. The purpose of this transformation is: lowering the entropy of a message (data). By lowering the entropy we reduce the information content.

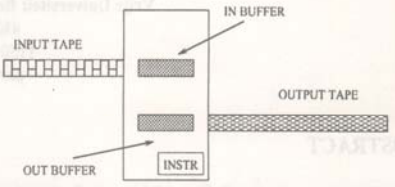


Figure 2: A simple virtual machine has been defined which bears some resemblance with an automaton. As a result it is possible to recode the message even more shortly (this is the domain of compression algorithms. . .).

The terminal and function set provided to the genetic programming software is designed to reduce the entropy, albeit under the good circumstances. It is up to the evolutionary pressure to bring forth those combinations of transformations that will reduce the entropy the most. Using the entropy we now can define a new condition for the transformation we wish to evolve:

$$\frac{H_{out} \times L_{out}}{H_{in} \times L_{in}} < 1 \tag{4}$$

Eq. (4) provides us with a computationally cheaper fitness function and is independent of any data compression algorithm. The formula requires the total information after transformation to be lower than before transformation. Note that we impose no limit on the length of the transformed data. Since we do not immediately compress the data, L_{out} can either be greater or equal to L_{in}.

4. SETUP

The setup used for this experiments will now be presented. A simple virtual machine has been defined which bears some resemblance with an automaton since it uses an input and an output tape. Data can be read from the input and new data can be put on the output (see figure 2).

Table 1: Short description of the different functions and terminals as well as their associated type(s).

(LOAD int data)) tape	**Load data**
(FWD data)) tape	Reload data
(REW data)) tape	Load output as data
(SEQ tape tape)) tape	Make sequences
(DIV int num)) int	Protected division
(DPCM num)) data	Difference code
(MIN num)) data	Average
(RAW)) data	Raw copy
(INV num)) data	Inversion
(PEC)) data	Pseudo exponent
(MTF)) data	Move to front

(SUB num)) data	Substitution
(ERC int)) int	Random constant
(END)) int	Length input
(ERC num)) num	Random constant

4.1 Strongly typed

The approach presented here uses a strongly typed genetic programming kernel written in C that produces LISP-like programs. The use of a typed genetic programming kernel is considered a must here since it allows structuring the programs, thus speeding up the search process. The different types and the instruction and terminal set used for this experiment will be presented shortly.
The instruction set can be divided into two catego Instructions that control the input tape and instructions process the data read from tape. Typing is used to structure programs that can be evolved. The first two types are int types, int and num. The num type represents small integer the range [0,20]. In order to integrate the instruction of machine two types were defined for the functions: tape data.

4.2 Settings

The experiments presented here were done using 2 populat of 500 individuals. The selection probabilities of the diffe genetic operations were: crossover 75%, mutation 20% reproduction 5%. The subpopulations exchanged 3 individuals every 10 generations. The exchanged individuals (the ones being exported) were selected using tournament selection. The imported ones replaced the worst individuals in the destination population. The exchange topology was a ring.

4.3 Platform

In this experiment our attention went toward the feasibility of this new preprocessing oriented approach.
The hardware used for this experiment a 8 Pentium II 350 Mhz cluster running Linux, the network used a 3Com 100Mbit/s non-blocking switch.

4.4 Input data

During these experiments the Canterbury Corpus has been used as well as various files ranging from bitmaps to word processor files. Meaning that the size of the input tape usually exceeded the order of several kilobytes and went even up to more than 1 megabyte in size. Some interesting results were observed.

5. RESULTS

To validate the results a set of software tools has been implemented. Meaning that next to the functions and terminals needed by the gp software, an encoding format has been designed and a decoder has been implemented. One can thus decode the preprocessed files and compare them with the original data with a program like diff.
The obtained reduction of the entropy is given in table 2. To demonstrate the usefulness of this work the preprocessed files have been compressed (see table 3).

5.1 Speedup

In order to compare the computational speedup the parallel and sequential runs have been timed. In figure 4 one can see that the computational speedup is linear (the parallel run time is half the sequential

Table 2: Comparison between the initial entropy and the entropy after preprocessing.

Filename	original H (bits)	new H (bits)
kennedy.xls	3.57	0.77
laptop.bmp	7.76	3.37
lena std.ppm	7.75	5.29
mosaic.pnm	7.78	4.15
peppers.ppm	7.66	5.35

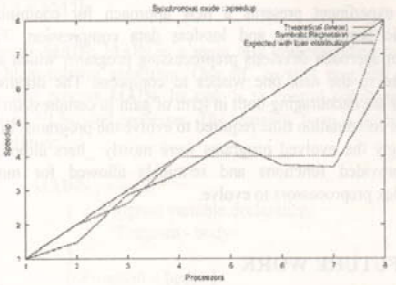


Figure 3: Speedup achieved on the standard symbolic regression problem in synchronous mode.

Time). This clearly illustrates the efficiency of the parallel kernel. One should keep in mind the fact that the user doesn't write a single line of parallel code! The solutions found in parallel and sequential mode are equivalent. This means that the search space exploration was the same.

5.2 Filters

Genetic programs tend to improve more slowly as their size increases. This is due to the increase of point's candidate for genetic change. In their experiments Nordin and Banzhaf[11] also had problems evolving programs that cover all the data. Therefore they have chosen to evolve separate programs for segments of fixed size, \chunking". This approach was not chosen although large amounts of data were used.
The solution that emerged here was filtering. With the introduction of the special END terminal in the set this unexpected result showed up. The entire data undergoes several transformations before

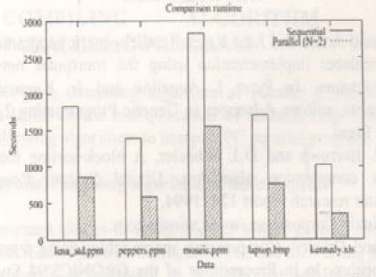


Figure 4: A comparison between the sequential and parallel execution time (2 nodes). 8 populations of 1000 individuals over 50 generations.

Being sent to the output, this filtering solution circumvents the data coverage problem that was initially encountered.

5.3 Introns

Furthermore during the experiments it was observed that by defining a more flexible virtual machine, which can cope with introns, the obtained programs were much better. The same observation was made by [9] but from another point of view. The introns combined with a relatively high mutation rate offer more opportunity for useful changes (i.e. new stage in the filters) to occur.

6. CONCLUSION

This experiment presents a new approach for combining genetic programming and lossless data compression. The chosen approach develops preprocessing programs which are tailored to the data one wishes to compress. The obtained results are encouraging both in term of gain in compression as for the computation time required to evolve the programs. Surprisingly the evolved programs were mostly _lters although the provided functions and terminals allowed for more complex preprocessors to evolve.

7. FUTURE WORK

When exchanges between populations are rather infrequent the load balancing seems to deteriorate.

Table 3: Difference in compression rate after preprocessing.

Filename	Bytes	bzip2-9 (GP)	bzip2 -9
kennedy.gp	1029744	24991	130280
laptop.gp	304182	131144	162270
lena std.gp	786466	538238	584538
mosaic.gp	910534	471795	658750
peppers.gp	786447	544211	635479

This occurs when the size of the programs being evolved by the different populations differs a lot. Further investigation is required to find out which parameters of the island model play a significant role for balancing the computational load.

REFERENCES

[1] David Andre and John Koza. Parallel genetic programming: A scalabel implementation using the transputer network architecture. In Peter J. Angeline and Jr. Kenneth E. Kinnear, editors, Advances in Genetic Programming 2. The Mit Press.

[2] M. Burrows and D.J. Wheeler. A block-sorting lossless data compression algorithm. Digital System Research Center research report 124, 1994.

[3] Winzip Corporation. www.winzip.com.

[4] Karel Driesen. Compressing sparse tables using a genetic algorithm. In In Proceedings of the GRONICS'94 Student Conference, Groningen, February 1994.

[5] I. De Falco et al. A kolmogorov complexity-base genetic programming tool for string compression. In Proceedings Seventh International Conference on Genetic Algorithms (ICGA97), 1997.

[6] H. Feiel and S. Ramakrishnan. A genetic approach to color image compression. In ACM 089791-850-9, 1997.

[7] Alex Fukunaga and Andre Stechert. Evolving nonlinear models for lossless image compression with genetic programming. In Genetic Programming 1998: Proceedings of the Third Annual Conference, pages 92{102, 1998.

[8] John R. Koza. Genetic Programming II. The MIT Press, 1994.

[9] Sean Luke. Code growth is not caused by introns. In Late Breaking Papers at the 2000 Genetic and Evolutionary Conference, 2000.

[10] Parallel Virtual Machine. http://www.netlib.org/pvm3/.

[11] Peter Nordin and Wolfgang Banzhaf. Programmatic compression of images and sound. In David B. Fogel John R. Koza, Davis E. Goldberg and Satnford University Rick L. Riolo ed.,editors, Proceedings of the First Annual Conference on Genetic Programming, pages 354{350. CA, USA, Mit Press.

[12] J. Parent. Parallel lil-gp technical report. Vrije Universiteit Brussel, http://parallel.vub.ac.be./~johan/Projects/.

[13] B. Punch and D. Zonker. lil-gp genetic programming version 1.1 beta. michigan state university, http://garage.cps.msu.edu/software/lilgp/index.html.

[14] William F. Punch. How e_ective are multiple populations in genetic programming. In K. Chellapilla K. Deb M. Dorigo D.B. Fogela M. Garzon D. Goldberg H. Iba J.R. Koza, W. Banzhaf and R.L. Riolo, editors, Genetic Programming 1998: Proceedings of the Third Annual conference. Morgan Kaufmann.

A Compiling Algorithm of Parallel C Program

Zhongyang Xiong
Department of Computer Science, Chongqing University
Chongqing, 400044, China
Email: zyxiong@cqu.edu.cn
And
Yufang Zhang
Department of Computer Science, Chongqing University
Chongqing, 400044, China
Email: zhangyf@cqu.edu.cn

ABSTRACT

This paper introduces a compiling algorithm of parallel C program, which designed for multiprocessor system. It focused on developing parallelism at sub-task level (program module level) during compiling procedure of C program. It can enhance parallelism and speed up the compiling of parallel C program.

Keywords: Parallel, C program, Compiling, Algorithm

1. INTRODUCTION

The rapid development of multiprocessor system provides an efficient way for enhancing the performance of computer system and puts forward more requirements for multiprocessor programming [1],[2]. To take advantage of multiprocessor parallelism, accelerate problem solving, it is imperative to research compiling technology of parallel program language. This paper studies compiling technology of parallel language based on multiprocessor and parallel C programming language.

Multiprocessor system in this paper has architecture of scalable ring. It is composed of 128 node processors and a host processor [5]. All processors are connected to public information belt, which has three 2*2 switch [4] components in order to reconfiguring the ring when ring has error. All processors communicate by cycle shift register that embedded in the belt. This architecture has the advantage of high transformation, non-conflict, simplicity, high reliability, flexibility and expandability.

2. PARALLEL C PROGRAM (PC)

As we know, C language is suited to system programming. The parallel language PC takes the C language as prototype. Based on it we develop parallel characteristic of PC program. The parallel language PC aims at multiple tasks disposing and develops parallelism at subtask (process or thread) level. The communication among tasks is implemented through message passing. Parallel language PC has a better synchronization characteristic, can ensure collaboration among processes or threads.

One PC parallel program consists of several independent executable program modules (processes or threads). Each process or thread runs in parallel mode, communicates each other to solve the giving problem. A whole PC parallel program defines two category processes: MAIN and PROCESS [5].

Process MAIN: MAIN is a main task corresponding to the problem-solving algorithm. One PC parallel program only has one process MAIN. Its task is to control collaboration among all parallel processes. The semantic form is described as follows:

```
<MAIN>:: = main
      {    Program variable declaration;
           <Program - body>
      }
      [<Function – body>]
<Program - body>:: = statement; |{statement;}
<Function - body>:: = function; |{function;}
```

In the above, "statement" is one of all legal C language statements or expansion PC parallel statements; "function" is a C language function and belongs to one process (process MAIN or PROCESS).

Process PROCESS: PROCESS is a subtask corresponding to parallel algorithm. One parallel program has several parallel executable PROCESSes, and each PROCESS has its own control flow, which has been distributed to one processor to execute by task scheduler. The semantic structure of PROCESS is:

```
<PROCESS>::= process_name [<formal parameter>]
      {Program variable declaration;
           <Program - body>
      }
      [<Function – body>]
```

3. COMPILING ALGORITHM OF PARALLEL C PROGRAM

We design a compiling algorithm of parallel C language, which based on the multiprocessor of scalable ring. It takes the following algorithms to compile PC parallel program.

3.1 Parallel compiling algorithm of main process

The primary idea is to recognize the MAIN process and all PROCESSes of parallel C program first, and then create some sub-processes respectively to compile the MAIN and all PROCESSes.

PROCEDURE main

```
(Parallel program "PC", executable code)
BEGIN
  /* Divide PC into a MAIN and n PROCESSes */
  Divide_process (PC, process pointer p ());
  for I = 1 to n+1 do
  /*Derives n+1 sub-processes "Sub_pcc(I)"*/
  /* To compile p(I) separately */
    Fork_sub_pcc(sub-process "Sub_pcc(I)",p (I) );
    /* Wake up scheduling process */
    Wakeup (Scheduler);
  L: sleep until waked up by sub-process "Sub_pcc (I)";
    /* Receive compiling result */
    Receive (Sub_pcc (I), result);
    Output (result);
    If not all parallel compiling sub-process ended then Go
      to L;
    Exit;                          /* this process end */
END
```

The function of Wakeup is to wake up scheduling process, to allocate processor and run compiling sub-processes. Receive is a receive communication primitive provided by multiprocessor operating system.

3.2 Recognition algorithm of parallel PROCESS

Divide_process recognizes all PROCESSes of parallel C program and disassembles them into several program modules. These modules are independent relatively and will be allocated to processors later. The definition of Divide_process is:

```
PROCEDURE Divide_process (PC, p( ))
BEGIN
    I = 0;
    Buf = 0;
    Word=" ";
    Read PC word to Word;
    Put Word into buffer Buf;
L2:  Read PC word to Word;
    /* Divide one "PROCESS" */
    if Word = "PROCESS" then
    BEGIN
      Take the program module from Buf
          and put into p(I);
      I++;
      Buf = 0;
      Put Word into buffer Buf;
      Go to L2
      /* Ready for dividing next "PROCESS" */
    END
    if Word = EOF then
      BEGIN
        Take the program module from Buf
            and put into p(I);
        Exit;          /* This process is ended */
      END
    Put Word into buffer Buf;
    Go to L2;
END
```

3.3 Deriving algorithm of parallel compiling sub-process

Fork_sub_pcc derives some compiling sub-process Sub_pcc( ) which compiles the MAIN or PROCESS on each processor concurrently. The procedure definition sees following description.

```
PROCEDURE
    Fork_sub_pcc( subprocess "Sub_pcc(I)", p(I) )
BEGIN
    Create sub-process "Sub_pcc(I)" of main;
    Add "Sub_pcc(I)" into ready queue SQ;
END
```

3.4 Parallel compiling algorithm of sub-process

The task of Sub_pcc (I) process is to compile program module p(I), and send the compiling result back to compiling main process. The description of algorithm is given below:

```
PROCEDURE Sub_pcc(I)
BEGIN
    Carry out parallel language preprocess program PCC
        for p(I);
    Compile p(I) through calling C language compiling
        program;
    /* Send compiling result back to main process */
    Send(main, result);
    Wakeup(main);
    Exit;
END
```

Send is a message sending primitive of multiprocessor operating system. The PCC program of PC deals with all parallel statements in p(I) in advance and converts them into C language.

3.5 Process scheduling algorithm

To suit with the characteristics of multiprocessor system and improve the performance of parallel compiling C program and resource usage, process scheduling adopts following policies:

(1) In general speaking, PROCESS always runs in node processor and MAIN runs in host processor.
(2) Same class process is scheduled by First Come First Service (FCFS) policy.

Process Scheduler dispatches all compiling sub-processes dynamically based on every processor's loading, in order to improve the system parallelism and resource usage maximally. The Scheduling procedure is described as follows.

```
PROCEDURE Scheduler
BEGIN
Loop: if SQ <> 0 then
      BEGIN
        /* Fetch the first process of ready queue*/
        RP = first(SQ);
          Choose the lightest loading processor NP based
            on status of each processor;
        Spread the RP to NP to run;
        Go to loop;
      /* Allocate processor for next process in SQ */
      END
      else
      BEGIN
        Sleep until waked up by process;
        Go to loop;
      END
END
```

4. PERFORMANCE STUDY

We assume that one parallel program PC is composed of one process MAIN and n processes (or threads) PROCESS. The time consuming of serial compiling of PC program is T_s, and T_p is time consuming of parallel compiling of PC. We analyze the performance of parallel compiling algorithm of PC program by computing T_s and T_p [3].

The serial compiling of PC is to compile the MAIN first, and then compile n PROCESSes one by one. Therefore, we have formulation Eq.(1)

$$T_s = T_{main} + T_{p1} + T_{p2} + \ldots + T_{pn} \quad (1)$$

T_{main} is time consuming for compiling MAIN process, T_{pi} is i^{th} PROCESS separately.

In the worst case:

$$T_{worst} = (n+1) * T_{max} \quad (2)$$
$$T_{max} = MAX\{T_{main}, T_{p1}, T_{p2}, \ldots, T_{pn}\}$$

The parallel compiling algorithm of PC program in this paper spreads MAIN and n PROCESSes into p processors to compiling simultaneously. Its time consuming is formulated as

$$T_p \le T_k + (n+1) * T_{max} * \alpha \div p \quad (3)$$

T_k is preprocessing overhead of PC and communication overhead during compiling. α is a ratio factor according to the processing ability of main processor and node processor, i. e. The processing ability of main processor is α times of node processor.

Not consider of T_k ($T_k << T_{max}$), for comparing formulation Eq. (2) and Eq. (3) we can get the result:

$$T_s / T_p = ((n+1) * T_{max}) / ((n+1) * T_{max} * \alpha / p) = p/\alpha \quad (4)$$

From formulation Eq. (4) we can conclude that parallel algorithm of PC has the accelerating effect obviously compared with serial compiling. The ideal acceleration ratio is p/α, we can improve it further by increasing processor number p, and this be confined by the ratio factor α. In the multiple processor system (128 processors)[5], the algorithm of parallel compiling can speeds up to 64 times.

5. CONCLUSION

The parallel compiling algorithm of PC program for multiprocessor develops parallelism at sub-task (program module). For developing parallelism of parallel language PC program to extensively, we should study the compiling procedure of C language extremely. The parallel compiling degree of PC program can be elevated through developing inner parallelism of C language compiling itself.

REFERENCES

[1] Kai Huang, Advanced Computer System Architecture, Tsinghua University and Guangxi Science Technology Press, 1999.10

[2] Kai Huang, Scalable Parallel Computing, China Machine Press, 1999.5

[3] W. Blume and R. Eigenmann, Performance Analysis of Parallelizing Compilers on Perfect Benchmarks Programs, IEEE Trans. On Parallel and Distributed System, 1992, vol.3, No.6, pp643-656

[4] Tong Fu, Cheng Daijie, Multiprocessor and Intelligent Multi-computer System, Chongqing University Press, 1988.2

[5] Tong Fu etc., The Multiprocessor System of Chongqing University, Technical Report, 1992

NOW-Based Distributed Implementation of Evolutionary Algorithms

Xiong Shengwu[1] Chu Wujun[2]
Institute of Computer Science and Technology, Wuhan University of Technology
Wuhan, Hubei, 430070,China
[1]Email: xiongsw@whapu.edu.cn
[2] Email: wjchu@263.net

ABSTRACT

This paper mainly discusses the application of MPI in distributed evolutionary computation on network of workstation. It also introduces how to develop a parallel evolutionary algorithms program in distributed computing environment based on MPI and how to exploit computing resource of network of workstations. A computing model to solve the problems of function optimization using island model distributed evolutionary algorithms is constructed to demonstrate the effectiveness of the proposed model.

Keywords: Distributed Evolutionary Algorithms, Message Passing Interface, Abstract Device Interface, Network of Workstations.

1. PARALLEL AND DISTRIBUTED EVOLUTIONARY ALGORITHMS AND SYSTEMS

Evolutionary algorithms (EAs) seek optimal or near-optimal solutions to hard search and learning problems by giving more chances of survival to filter individuals in an evolving population in which each individual represents a feasible solution to the given problem through a suitably coded string of symbols. Evolutionary algorithms have found increasing application to many problems in diverse areas such as hard function and combinatorial optimization, neural nets evolution, routing, planning and scheduling, management and economics, machine learning and robotics and pattern recognition [1]. The search space in genetic programming is the space of all computer programs composed of functions and terminals appropriate to the problem domain. Suitable functions and terminals are determined for the problem at hand and an initial random population of trees is constructed. The population then evolves with fitness being associated to the actual execution of the program and with genetic operators adapted to the tree representation. These procedures need lots of time. Evolutionary algorithms are well suited to parallel implementation [2]. In computer runs of evolutionary algorithms for most categories of problems, relatively little computer time is spent on the one-time task of creating the initial population at the beginning of the run and relatively little computer time is spent on the execution of the Darwinian selection and genetic operations on each generation during the run. Instead, the task of measuring the fitness of each individual in each generation of the evolving population is (for most categories of problems) the dominant component of the computational burden (with respect to computer time) in solving non-trivial problems using evolutionary algorithms. These observations give rise to the most commonly used approach to parallelization of evolutionary algorithms.

The most important advantage of PEAs is that in many cases they provide better performance than single population-based algorithms, even when the parallelism is simulated on conventional machines. The reason is that multiple populations permit speciation, a process by which different populations evolve in different directions (i.e. toward different optima). For this reason Parallel EAs are not only an extension of the traditional EA sequential model, but they represent a new class of algorithms in that they search the space of solutions differently. Interestingly, PEAs often allow theoretical analyses, which are not harder than those for sequential EAs. Owing to the large number of model proposed in the literature, the only problem that one has to face to use PEAs is how to determine which parallel model to use. The most popular parallel models are the fine-grained or grid models, and the coarse-grain or island models. The advantage of parallel EA for difficult problems is that they can handle larger populations in reasonable times and favor cooperatively in the search for good solutions.

Parallel processing systems currently used for processing EAs have undergone their own evolution from sequential systems, through shared-memory parallel systems until message-passing or distributed-memory systems. Corresponding computational models, which also were changing, influenced a possible method of implementing EAs. Early parallel processing systems were shared-memory systems, in which all processors communicated by reading and writing to memory locations in a common address space. In the simplest case such systems, classified as SIMD (Single Instruction Multiple Data) machines, enable the execution of one common program by all processors but using different data. Shared-memory systems offer a possibility of easy parallel implementation of algorithms, which, however, is limited by mechanisms of synchronization of memory access. To provide massive parallelism for algorithms, distributed-memory systems were designed. A distributed-memory multiprocessor is a collection of processors interconnected by some communication network. Processors communicate sending messages through the network; they do not share any common memory. In such systems, classified as MIMD (Multiple Instructions Multiple Data) machines, each processor may execute its own program using its own data. Another representative of distributed-memory systems is a NOW (network of workstation) machines, which unlike to multiprocessors are geographically dispersed systems occupying some limited local area. It is worth mentioning that in distributed-memory systems, which basically are message-passing systems, a shared-memory model can be also implemented.

Increases in computing power can be realized in two ways: either by using a faster computer or by parallelizing the application. The first approach is aided by the fact that computer speeds have approximately doubled every 18 months in accordance with Moore's law and are expected to continue to do so. The second approach (i.e., parallelization) is available for applications that can be parallelized efficiently using message-passing system, such as MPI. Evolutionary algorithms, genetic programming, and other techniques of evolutionary computation are highly amenable to parallelization.

One of the most common parallel computing environments is NOW (network of workstations). It is popularly used Ethernet-connected personal workstations as a free computational resource .In many cases, the workstation collection includes machines from multiple vendors. Interoperability is provided by the TCP/IP standard. MPICH runs on workstations from Sun (both SunOS and Solaris), DEC, Hewlett-Packard, SGI, and IBM. And now, 6the Intel 486 and Pentium compatible machines have been able to join the Unix workstation family by running one of the common free implementations of Unix, such as FreeBSD, NetBSD, or Linux. One way to simulate a parallel environment on the workstations is to use a software system called the Message Passing Interface (MPI).

MPI is intended to address only message passing and to build on the most useful concepts and constructs in the current message passing systems [3]. These are primarily covered in the areas of point-to-point message passing, collective communications, process groups and topologies. Even commercial implementations of MPI are available now. MPI is a specification for a standard library for message passing that was designed by the MPI Forum, its goal is to provide users with a free, high-performance implementation on a diversity of platforms, allows workstations to operate together as one machine. This cooperation gives the appearance of a large parallel machine. MPICH is portable and high-performance implementation of MPI. It is structured in terms of an abstract device interface (ADI). The ADI defines low-level communication-related functions that can be implemented in different machines. MPICH runs on all of these workstations and on heterogeneous collections of them .An important family of non-Unix operating systems is supported by Microsoft. MPICH has even been ported to Windows .The MPICH specification was designed to allow high performance in the sense that semantic restrictions on optimization were avoided wherever user convenience would not be severely impacted. Furthermore, a number of features were added to enable users to take advantage of optimizations that some systems offered, without acting portability to other systems that did not have such optimizations available.

2. THE DISTRIBUTED IMPLEMENTATION OF FUNCTION OPTIMIZATION PROBLEM

Two research groups, reporting jointly in this communication, prepared class libraries with distinct design factors and features, in order to address this situation: MPI++, designed at Mississippi State University, and Object-Oriented MPI (OOMPI), designed by the University of Notre Dame. Both MPI++ and OOMPI were explicitly designed for inheritance; users can derive their own objects from either class library. MPI++ uses several features of C++ that are absent from the subsequently accepted MPI-2 C++ bindings, while remaining faithful to the majority of function signatures from the MPI C bindings. OOMPI emphasizes object-oriented design and ease of use rather than compliance with the MPI C bindings. In addition to reference and const semantics, OOMPI makes extensive use of default arguments, overloaded function names, and inheritance. Both libraries are of interest because they proceeded C++ bindings that were later added to MPI and significantly in influenced these standardized extensions.

In particular, the MPI C++ bindings were accepted to be a minimalistic approach to the message-passing model. A small set of objects is provided which encapsulate all MPI data and functionality. While much of the C function signatures are preserved, the C++ bindings take advantage of several inherent features of the C++ language, to include reference and const semantics. More advanced features of C++, such as overloading and polymorphism, were not used in order preserve a direct and unambiguous mapping to the specified functionality of MPI. While the design criteria appear restrictive, the MPI C++ bindings provide both a simple object-oriented model that programmers can immediately use in their C++ programs as well as a sound basis for building class libraries that use more advanced features of C++.

The way in which PGA (parallel genetic algorithm) can be implemented depends on the following elements:
1) How fitness is evaluated and mutation is applied
2) If single or multiple subpopulations (demes) are used
3) If multiple populations are used, how individuals are exchanged
4) How selection is applied (globally or locally)

Try to minimize Ackley function, we simply use parallel evolutionary algorithms to solve it and make a convenience of MPICH to exploit parallelism of EAs to speed up computation and reach global optima. .

A run of genetic programming begins with the initial creation of individuals for the population. Then, on each generation of the run, the fitness of each individual in the population is evaluated. Then, on each generation, individuals are selected (probabilistically based on fitness) to participate in the genetic operations (e.g., reproduction, crossover, mutation, and architecture-altering operations). These three steps (i.e., fitness evaluation, Darwinian selection, and genetic operations) are iteratively performed over many generations until the termination criterion for the run is satisfied. Typically, the best single individual obtained during the run is designated as the result of the run.

Here, we use abstract data type to describe computing model based on the parallel evolutionary algorithms.

```
Population
Data member
int POPSIZE;//size of population
double   PCROSSOVER;          //possibility of crossover
double PMUTATION;             //possibility of mutation
double Upper_bound;           //upper bound of variable
double Lower_bound;           //lower bound of variable
int g;                        //current generation
Seqlist<genetype *>genes;
int nVars;                    //quantity of variables
Construction operation: initate a population
population(int/*popsize*/SIZE,
        double/*possibilityofmutation*/pm,
```

*double/*possibilityofcrossover*/pc,*
double / upper bound*/u,*
*double/*lower bound*/ l,*
*int/*quantity of variables */ n_vars);*
Destruction operation: ~population();
Release resource.
Evaluation operation: Evaluate()
Calculate fitness
Import operation: void Import(individual &)
Import the better individuals from another population

Keep the best operation: keep_thebest();
Keep track of the best individual in current population
Migration operation: void Migratet(individual &)
Export the best individual to neighbor population.
Select operation: void Select()
Standard proportional selection for maximum/minimum problems incorporating elitist model, which make sure best individual survives.
Elitist operation: void Elitist()
The best member of the previous generation is stored as the last in the list of individual.
Crossover operation: void Crossover ()
Select two parents that take part in the crossover.
Mutation operation: void Mutate ()
Select a individual to do mutation.
Report operation: void Report()
Output the current population.

In the "island" approach to parallelization of genetic programming, the population for a given run is divided into semi-isolated subpopulations (called *demes*). Each subpopulation is assigned to a separate processor of the parallel computing system. The run begins with the one-time random creation of a separate population of individuals at each processor of the parallel computer system. This process of initial random creation takes place in parallel at each processor. As soon as each separate processor finishes this one-time task, it begins the main generational loop.

In the main generational loop of genetic programming, the task of measuring the fitness of each individual is first performed locally at each processor. Then, the Darwinian selection step is performed locally at each processor. In the evolution procedure, each subpopulation sends its elite individual to its neighbor. If the elite individual better than local elite individual, then its neighbor accepts it and replaces the local one. Finally, the genetic operations are performed locally at each processor. The processors operate asynchronously in the sense that each generation starts and ends independently at each processor. Because each of these tasks is performed independently at each processor and because the processors are not synchronized, this asynchronous island approach to parallelization efficiently uses all the processing power of each processor. These can be done upon MPICH.

Upon completion of a generation (or other interval), the best individual in each subpopulation are probabilistically selected (based on fitness) for emigration from each processor to various neighboring processors. The processors are typically arranged in a rectangular toroidal topology. Emigrants are exported to the neighboring processors, where they wait in an "importer" buffer of the destination processor until the destination processor is ready to assimilate its accumulating immigrants. This assimilation typically occurs just after the processor has exported its own emigrants at the end of its generation. The immigrants are typically inserted into the subpopulation at the destination processor in lieu of the just-departed emigrants of that processor's subpopulation. The overall iterative process proceeds asynchronously from generation to generation on each separate processor.

The inter-processor communication requirements of migration are low because only a modest number of individuals migrate during each generation and because each migration is separated by substantially longer periods of time for fitness evaluation, Darwinian selection, and genetic operations.

Ackley function is an experimental function. It is modulated by exponent function that was overlapped by properly magnified cosine wave. Its distinctive feature is too many punch hole formed by modulated cosine wave in an almost flat area. So it's curved surface looks out of level. Ackley function can be stated as follows:

$$\min f(x_1,x_2) = -c_1 \cdot \exp\left(-c_2 \cdot \sqrt{\frac{\sum_{j=1}^{2} x_j^2}{n}}\right) - \exp\left(\frac{1}{n}\sum_{j=1}^{2}(c_3 \cdot x_j)\right) + c_1 + e$$

Subject to: $-5 \le x_j \le 5; j = 1,2$

$c_1 = 20, c = 0.2, c_3 = 2\pi, e = 2.71282$

Ackley had pointed out that search of this function was difficult because the methodically local optimal algorithms inevitably trap in local optima in mountain climbing procedure [4]. But if we explore larger neighborhood, we can travel over the obstruct valley.

①Real number encoding.

$v_i = [x_1, x_2],\ x_i = Random(-5,5), i = 1,2$

②Design of evaluation function.

$Evaluate(f) = -f(x_1, x_2)$;

③Select strategy. In order to avoid premature phenomenon, use the Sigma proportional manipulation technique to translate individual fitness.

If $\sigma(t) > 0$, $ExpVal(i) = 1 + (f(i) - f(t))/2\sigma(t)$

If $\sigma(t) = 0$,

$ExpVal\ (i) = 1$,

$\sigma(t)$ is standard variance of the t generation population. Then apply the select strategy based on proportion to $ExpVal\ (i)$

④ Mathematic crossover.

$v_1 = v_1 + (1-\lambda)v_2; v_2 = v_2 + (1-\lambda)v_1;\ \lambda \in (0,1)$;

⑤ Non-uniform mutate. To a selected parent v, if his element x_k is selected to mutate, his successor is

$v' = [x_1, \cdots, x_k', \cdots, x_n]$,

$x_k' = x_k + \Delta(t, x_k^U - x_k)$

or $x_k' = x_k - \Delta(t, x_k - x_k^L)$,

$\Delta(t,y) = y \cdot r\left(1 - \frac{t}{T}\right)^b$

Subject to: x_k^U is upper bound of x_k, x_k^L is lower bound of x_k

$r = Random(0,1)$;

T number of maximum generation,

b parameter decided the nonuniform degree.

⑥ Using elitist model. In each evolve loop, keep the track of best individual (whose fitness is the biggest) of current population and store it in somewhere. In next generation, if the best individual were worse than the best individual of the previous generation, the latter one would replace the worst individual of the current population.

⑦ Communication .Here we use ring topology. If the immigrant from neighbor better than the best of current population, the replace the latter one, otherwise, refuse it.

In PEA implementation of optimization of Ackley function, three P II 266 computers participate in parallel computation,. And we also do sequential EA implementation on personal workstation, so we can analyze the efficiency of PGA compare with traditional GA. Control parameters and final solution is shown as follows:

	PEA	EA
Pop_size	10	10
P_m	0.1	0.1
P_c	0.3	0.3
Solution	(1.597e-016, 1.483e-016)	(1.617e-017, 1.422e-016)
Generation	278	797

3. CONCLUSI ON

From the experimental results, we can see the most important advantage of PEAs is that in many cases they provide better performance than single population-based algorithms, even when the parallelism is simulated on conventional machines. In this paper, we discuss the problem of optimization of Ackley function implemented by PEA based MPI. MPI has been an industry standard, we can use it to exploit the computing resource in NOW. And use it to accomplish parallel computation, solve big and complex problems. The next we want to do is to use PEA based MPI to solve the traditional NP difficult problems, such as MST (Minimal Spanning Trees), SPP (Set Partition Problem). Furthermore, use the MPE (MPI Extension routine lib) to implement graphical interface of parallel computation, and design the topologies of NOW to gear to different requirements.

REFERENCES

[1] Bauer, R., Genetic Algorithms and Investment Strategies, New York: John Wiley & Sons, 1994.

[2] Zbigniew Michalewicz Genetic Algorithms + Data Structures=Evolution Programs, Berlin: Springer-Verlag, 1994.

[3] Message Passing Interface Forum. MPI: A message-passing interface standard. International Journal of Supercomputer Applications, 8(3/4): 165-414,1994.

[4] Ackley D., A Connectionist Machine for Genetic Hillclimbing, Boston: Kluwer Academic Public Publishers, 1987.

Using Genetic Algorithm to Solve Fuzzy Flow-shop Scheduling Problems with Fuzzy Processing Time and Fuzzy Due Date

Zhaoqiang Geng, Yiren Zou
ERC of Integrated Automatic Technology
Institute of Automation, Chinese Academy of Sciences
Beijing, 100080, P.R. China, Email: gzq@sample.ia.ac.cn

ABSTRACT

This paper considers two kinds of fuzzy flow-shop scheduling problems with fuzzy processing time and fuzzy due date. In the first kind of problem, fuzzy processing time is denoted by a triangular fuzzy number and fuzzy due date is denoted by a trapezoid fuzzy number. As for the second kind of problem, fuzzy processing time is denoted by a trapezoid fuzzy number and fuzzy due date is denoted by a 6-point fuzzy number. On the basis of the agreement index of fuzzy due date and fuzzy completion time, the model is formulated. The maximum average agreement index is used as optimized object. Besides the above two kinds of problems with both fuzzy processing time and fuzzy due date, we also study the problems with only fuzzy processing time where the minimum fuzzy completion time is taken as optimized object. We adopt genetic algorithm to find the optimal sequencing of the given problem. Two numerical examples are shown to illustrate the effectiveness and feasibility of our proposed algorithm

Keywords: Flow-Shop Scheduling, Fuzzy Processing Time, Fuzzy Due Date, Agreement Index, Genetic Algorithm

1. INTRODUCTION

The flow shop scheduling problem has aroused many scholars' attention since it was first refer by Johnson in 1954 [1]. Most research work focus on certain flow shop scheduling problems where the job's processing time and due date are considered as certain value. However, when studying flow shop scheduling problems in the real-world situations when some uncertain factors are incorporated into the problems, it's very difficult to get the certain value of processing time and due date. In such conditions, it may be more appropriate to consider fuzzy processing time due to various factors and fuzzy due date tolerating a certain amount of earliness or tardiness in the due date. We call this kind of flow shop scheduling problem as fuzzy flow shop scheduling problem (FFSSP).

There is not a simple method to solve FFSSP at present. Integer programming and branch- and-bound techniques are two common methods. While they are not very effective when solving large-scale problem, even medium scale one. Intelligent optimized techniques, including genetic algorithm, simulating anneal, tabu research and artificial neural network, have developed rapidly in recent years. Among all, genetic algorithm is especially outstanding because of its excellent computing capacity and application effect.

This paper uses genetic algorithm to search the optimal sequencing of FFSSP.

2. MATHEMATICAL MODEL

In general, an n × m flow shop scheduling problem is formulated as follows. Let n jobs be processed on m machines, and each job needs m operations, each operation needs different machine. It is assumed here that only one operation can be processed on each machine at a time, and preemption is not allowed. Each job has the same processing sequence. The fuzzy processing time of job i on machine j is represented by $\tilde{p}_{ij}$ (i=1,2 , ··· n ; j=1,2, ··· m). The objective is to find the processing sequence of jobs to maximize a certain performance index [2].

In this paper, according to different problems, the fuzzy processing time $\tilde{p}_{ij}$ may be denoted by a triangular fuzzy number $(p_{ij}^1, p_{ij}^2, p_{ij}^3)$ as shown in Fig.1 or a trapezoid fuzzy number $(p_{ij}^1, p_{ij}^2, p_{ij}^3, p_{ij}^4)$ as shown in Fig.2. The fuzzy due date is represented by the degree of satisfaction with respect to the job completion time, and can be denoted by a trapezoid fuzzy number $(d_i^c, d_i^a, d_i^b, d_i^d)$ as shown in Fig.3 or 6-point fuzzy number $(d_i^e, d_i^c, d_i^a, d_i^b, d_i^d, d_i^f)$ as shown in Fig.4.

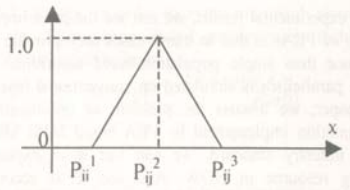


Fig.1 Fuzzy processing time (triangular fuzzy number)

The membership function of fuzzy processing time denoted by triangular fuzzy number is defined as:

$$\mu_{ij}(x)=\begin{cases} 0 & x \le p_{ij}^1, x \ge p_{ij}^3 \\ \dfrac{x-p_{ij}^1}{p_{ij}^2-p_{ij}^1} & p_{ij}^1 < x \le p_{ij}^2 \\ \dfrac{p_{ij}^3-x}{p_{ij}^3-p_{ij}^2} & p_{ij}^2 < x \le p_{ij}^3 \end{cases}$$

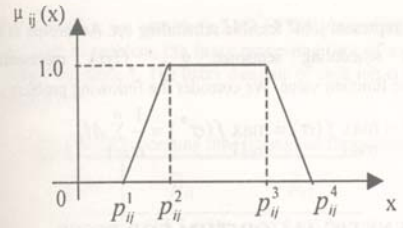


Fig.2 Fuzzy processing time (trapezoid fuzzy number)

The membership function of fuzzy processing time denoted by trapezoid fuzzy number is defined as

$$\mu_{ij}(x)=\begin{cases}1 & p_{ij}^2\le x\le p_{ij}^3\\ \dfrac{x-p_{ij}^1}{p_{ij}^2-p_{ij}^1} & p_{ij}^1\le x<p_{ij}^2\\ \dfrac{p_{ij}^4-x}{p_{ij}^4-p_{ij}^3} & p_{ij}^3\le x<p_{ij}^4\\ 0 & x<p_{ij}^1, x>p_{ij}^4\end{cases}$$

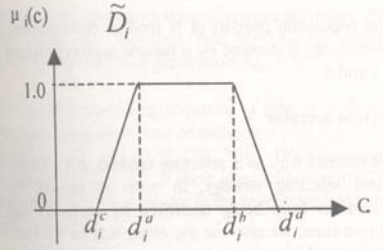


Fig.3 Fuzzy due date (trapezoid fuzzy number)

The membership function of fuzzy due date denoted by trapezoid fuzzy number is defined as

$$\mu_i(c)=\begin{cases}1 & d_i^a\le c\le d_i^b\\ \dfrac{c-d_i^E}{d_i^a-d_i^E} & d_i^E\le c<d_i^a\\ \dfrac{d_i^M-c}{d_i^M-d_i^b} & d_i^b\le c<d_i^M\\ 0 & c<d_i^E, c>d_i^M\end{cases}$$

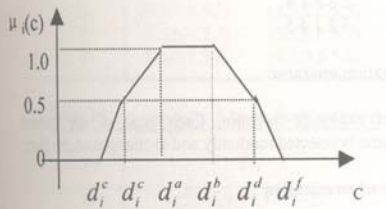


Fig.4 Fuzzy due date (6-point fuzzy number)

The membership function of fuzzy due date denoted by 6-point fuzzy number is defined as

$$\mu_i(c)=\begin{cases}1 & d_i^a<c<d_i^b\\ \dfrac{1}{2}+\dfrac{1}{2}\dfrac{c-d_i^c}{d_i^a-d_i^c} & d_i^c<c<d_i^a\\ \dfrac{1}{2}\dfrac{c-d_i^e}{d_i^c-d_i^e} & d_i^e<c<d_i^c\\ \dfrac{1}{2}+\dfrac{1}{2}\dfrac{d_i^d-c}{d_i^d-d_i^b} & d_i^b<c<d_i^d\\ \dfrac{1}{2}\dfrac{d_i^f-c}{d_i^f-d_i^d} & d_i^d<c<d_i^f\\ 0 & c<d_i^e, c>d_i^f\end{cases}$$

We formulate FFSSP model considering both fuzzy processing time and fuzzy due date which is seldom hit in previous studies. Now we give the detailed description of the mathematical model.

Let $\tilde{C}(j_i,k)$ represent fuzzy completion time of job j_i on machine k. $\{j_1,j_2,\cdots,j_n\}$ shows a feasible scheduling sequence, then the fuzzy completion times of n×m FFSSP are given as follows.

$$\tilde{C}(j_1,1)=\tilde{p}_{j_1 1} \quad (1)$$

$$\tilde{C}(j_1,k)=\tilde{C}(j_1,k-1)+\tilde{p}_{j_1 k};\quad k=2,\cdots m \quad (2)$$

$$\tilde{C}(j_i,1)=\tilde{C}(j_{i-1},1)+\tilde{p}_{j_i 1};\quad i=2,\cdots n \quad (3)$$

$$\tilde{C}(j_i,k)=\max\{\tilde{C}(j_{i-1},k);\tilde{C}(j_i,k-1)\}+\tilde{p}_{j_i k};\quad i=2,\cdots,n;k=2,\cdots m \quad (4)$$

The maximum fuzzy flow time is

$$\tilde{C}_{\max}=\tilde{C}(j_n,m) \quad (5)$$

From equation (1) to equation (5), we need use the adding and maximizing operation of two fuzzy numbers. We also know the ranking method of comparing fuzzy numbers.

Here, we use fuzzy addition operator [3]. For example, if $\tilde{A}=(a^1,a^2,a^3)$, $\tilde{B}=(b^1,b^2,b^3)$, $\tilde{C}=(c^1,c^2,c^3,c^4)$, $\tilde{D}=(d^1,d^2,d^3,d^4)$ then

$$\tilde{A}+\tilde{B}=(a^1,a^2,a^3)+(b^1,b^2,b^3)$$
$$=(a^1+b^1,a^2+b^2,a^3+b^3).$$
$$\tilde{C}+\tilde{D}=(c^1,c^2,c^3,c^4)+(d^1,d^2,d^3,d^4)$$
$$=(c^1+d^1,c^2+d^2,c^3+d^3,c^4+d^4).$$

For the maximizing operation of two fuzzy numbers, we use the following formula.

$$\max(\tilde{A},\tilde{B})=(\max(a^1,b^1),\max(a^2,b^2),\max(a^3,b^3))$$
$$\max(\tilde{C},\tilde{D})=(\max(c^1,d^1),\max(c^2,d^2),$$
$$\max(c^3,d^3),\max(c^4,d^4))$$

In FFSSP, when comparing two fuzzy numbers, some ranking methods become necessary. In this paper, we adopt the following ranking method for triangular fuzzy numbers [4].

Criterion 1. $C_1(\tilde{A})=\dfrac{a^1+2a^2+a^3}{4}$

If $C_1(\tilde{A})>C_1(\tilde{B})$, then $\tilde{A}>\tilde{B}$.

Criterion 2. $C_2(\tilde{A})=a^2$

If $C_1(\tilde{A})=C_1(\tilde{B})$, $C_2(\tilde{A})>C_2(\tilde{B})$,

then $\tilde{A}>\tilde{B}$.

Criterion 3. $C_3(\tilde{A})=a^3-a^1$,

If $C_1(\tilde{A})=C_1(\tilde{B})$, $C_2(\tilde{A})=C_2(\tilde{B})$,

$C_3(\tilde{A})>C_3(\tilde{B})$, then $\tilde{A}>\tilde{B}$.

When the fuzzy completion times are trapezoid fuzzy numbers, we adopt the following ranking method to compare them. The method is proposed for comparing fuzzy numbers based on the compensation of the areas determined by the membership functions [5]. This can be shown in Fig.5.

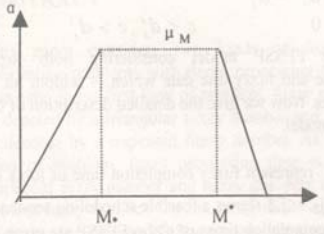


Fig.5 The mean value (in dotted lines) of a fuzzy number $\tilde{M}$ (in continuous lines)

We choose the gravity center of this interval [M*,M^*] (M*+M^*)/2 as a criterion to compare two trapezoid fuzzy numbers.

Criterion $C_1(\tilde{M})=\dfrac{M_* + M^*}{2}$

If $C_1(\tilde{M})>C_1(\tilde{N})$, then $\tilde{M}>\tilde{N}$.

For the fuzzy completion time for each job expressed as a triangular fuzzy number of trapezoid $\tilde{C}_i$, as an index showing the portion of $\tilde{C}_i$ that meets the fuzzy due date $\tilde{D}_i$, it is convenient to adopt the agreement index (AI) of the fuzzy completion time $\tilde{C}_i$ with respect to the fuzzy due date $\tilde{D}_i$. Here, we define AI as value of the area ▲ of membership function intersection divided by the area △ of the $\tilde{C}_i$ membership function [4]. This can be shown in Fig.6 and Fig.7. To be more explicit, AI can be expressed as follow.

$AI=(area\tilde{C}_i \cap \tilde{D}_i)/(area\tilde{C}_i)$.

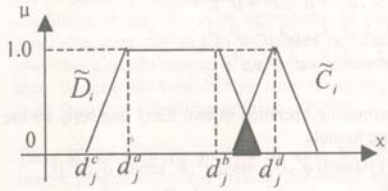


Fig.6 Agreement Index (Ⅰ)

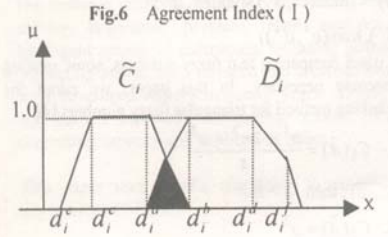


Fig.7 Agreement Index (Ⅱ)

Let Π represent jobs' feasible scheduling set. According to a specific scheduling sequence σ, $f(\sigma)$ represents objective function value. We consider the following problem

$$\max_{\sigma\in\Pi} Z=\max_{\sigma\in\Pi} f(\sigma)=\max_{\sigma\in\Pi} f(\sigma^*)=\frac{1}{n}\sum_{i=1}^{n} AI_i$$

3. GENETIC ALGORITHM FOR FFSSP

Genetic algorithm was first proposed by Professor J.Holland in Michigan University in 1975 and has been used widely in solving scheduling problems [6]. We use it to search the optimal sequence. Now we give the detailed design of our genetic algorithm.

3.1 Design of coding

The chromosome is denoted by jobs' processing sequence. This is a natural expressing way of scheduling problems. For example, 1 3 4 5 2 means the processing sequence of jobs is j_1, j_3, j_4, j_5, j_2.

3.2 Method for generating initial population

The initial population consists of N random chromosomes. Each chromosome is denoted by n natural numbers that are between 1 and n.

3.3 Selection operator

We adopt roulette way as a selecting method. It's a fitness proportional selecting strategy. In order to prevent the optimal solution from being destroyed by crossover and mutation operation, we also use the elitist way on the basis of roulette way, which can accelerate the searching speed.

3.4 Crossover operator

We adopt one-point crossover. First, select a breakpoint randomly. Second, select the part behind of the breakpoint of the first parent chromosome as a part of the child chromosome. Finally, select legal genes of the second chromosome to fill the remnant part of the child chromosome. Legal genes are different from the selected genes of the first chromosome. By these operations, we can avoid producing illegal child chromosome. For example,

Parent 1 1 2 3 4 5
Parent 2 3 5 2 1 4
Suppose the breakpoint is 3, then
Child 1 2 3 5 1 4
Child 2 3 2 1 4 5

3.5 Mutation operator

We adopt exchange operator. Each gene of the parent chromosome is selected randomly and exchanged each other.

3.6 Rule of termination

Use the preset maximum generation N_{max} as termination condition.

4. NUMBERICAL EXAMPLES

For simplicity, we consider a 5×2 FFSSP. According to the first kind of problem, the fuzzy processing time of each job is shown in table 1. The fuzzy due date of each job is shown in table 2.

Table 1 fuzzy processing time (triangular fuzzy number)

工件i	$\tilde{P}_{i1}$	$\tilde{P}_{i2}$
1	(2.5,3,3.5)	(5.5,6,6.5)
2	(4.5,5,5.5)	(1.5,2,2.5)
3	(0.5,1,1.5)	(1.5,2,2.5)
4	(5.5,6,6.5)	(5.5,6,6.5)
5	(6.5,7,7.5)	(4.5,5,5.5)

Table 2 fuzzy due date (trapezoid fuzzy number)

jobi	$\tilde{D}_i$
1	(8,9,10,11)
2	(9.5,11,12.5,14)
3	(8,10,13,15)
4	(21,23.5,26,28.5)
5	(20,23,24,27)

We give the parameter values of genetic algorithm. Population size N=20, the crossover rate P_c=0.8, the mutation rate P_m=0.05, the maximum generation N_{max}=100.

The optimal processing sequence of jobs is 1—2—3—4—5, the fuzzy completion time of each job is (8,9,10), (9.5,11,12.5), (11,13,15), (18.5,21,23.5), (24,27,30). The maximum average agreement index is 0.7. If there is no requirement of due date, that means we only consider fuzzy processing time, we got the following result.

The optimal processing sequence of jobs is 1—3—4—5—2, the fuzzy completion time of each job is (8,9,10), (9.5,11,12.5), (15,17,19), (19.5,22,24.5), (21,24,27). The minimum fuzzy completion time is (21,24,27).

As for the second kind of problem, the fuzzy processing time of each job is shown in table 3. The fuzzy due date of each job is shown in table 4.

Table 3. fuzzy processing time (trapezoid fuzzy number)

工件i	$\tilde{P}_{i1}$	$\tilde{P}_{i2}$
1	(2.5,3,3.5,4)	(5.5,6,6.5,7)
2	(4.5,5,5.5,6)	(1.5,2,2.5,3)
3	(0.5,1,1.5,2)	(1.5,2,2.5,3)
4	(5.5,6,6.5,7)	(5.5,6,6.5,7)
5	(6.5,7,7.5,8)	(4.5,5,5.5,6)

Table 4 fuzzy due date (6-point fuzzy number)

jobi	$\tilde{D}_i$
1	(7,7.5,9,10,11.5,12)
2	(12.5,13.25,15,16,17.75,18.5)
3	(17,18,20,21,23.24)
4	(13,14.25,16,18,19.75,21)
5	(21,22,24,33,35,36)

The parameter values of genetic algorithm are the same as those in the first kind of problem.

The optimal processing sequence of jobs is 1—2—3—4—5, the fuzzy completion time of each job is (8,9,10,11), (9.5,11,12.5,14), (11,13,15,17), (18.5,21,23.5,26), (24,27,30,33). The maximum average agreement index is 0.45.

Likewise, if there is no requirement of due date, that means we only consider fuzzy processing time, we got the following result.

The optimal processing sequence of jobs is 1—3—4—5—2, the fuzzy completion time of each job is (8,9,10,11), (9.5,11,12.5,14), (15,17,19,21), (19.5,22,24.5,27), (21,24,27,30). The minimum fuzzy completion time is (21,24,27,30).

5. CONCLUSIONS

This paper made a deep study of fuzzy flow shop scheduling problem. We adopt fuzzy set theory to formulate a FFSSP model that considers fuzzy processing time and fuzzy due date. We take average agreement index of n jobs as optimized object and use genetic algorithm to search the optimal sequence. We also study the FFSSP with only fuzzy processing time. Finally, We give numerical examples to illustrate the effectiveness of our proposed method which provides a new way of studying planning and scheduling problems in fuzzy circumstances.

REFERENCES

[1]Johnson, S. Optimal two-and-three stage production schedules with setup times included. Naval Research Logistics Quartely. 1, pp.61~68,1954.

[2]Wang Ding-wei, Tang Jia-fu, Huang Min. Genetic algorithm and engineering design. Beijing: Science Press, 2000.

[3]Li Fan, Fuzzy information processing sysytem, Beijing, Beijing University Press, 1998.

[4]Masatoshi Sakawa, Ryo Kubota. Fuzzy programming for multi-objective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms. European Journal of Operational Research. Vol.120, pp.393~407,2000.

[5]P. Fortemps and M.Roubens, 'Ranking and defuzzification methods based on area compensation," Fuzzy Sets and Systems, Vol..82, pp.319~330, 1996.

[6]Guo-Liang Cheng and Xu-Fa Wang, Genetic Algorithm and Its Applications, Beijing, BJ: People's Post and Telecommunication Press, 1999

Convergence of Internet and Digital Television: Challenges and Achievements

Yakup Paker
Queen Mary, University of London
Mile End Road, London E1 4NS, UK
paker@dcs.qmul.ac.uk

ABSTRACT

This paper describes a merging trend of digital TV and PCs. The Mpeg-4 and Mpeg-7 standards as well as some protocols have been discussed in details.

Keywords: Internet, Digital Television, Mpeg 4, Mpeg 7, Protocol.

1. INTRODUCTION

Since the British Broadcasting Corporation (BBC) started the first regular TV service for home viewing in 1936, TV has become a universal medium for news, information and entertainment, taking up considerable time in an individual's daily life with much impact on the cultural and political life of most countries. Thus after half a century since its inception the importance of TV for individuals and countries can not be over emphasised. The technology of TV delivery has changed surprisingly little over the years, the biggest changes coming from the introduction of colour in the fifties and satellite in the eighties. A new technology is taking over the TV, however, called digital TV. In Europe, alongside the analogue channels, the digital TV is being introduced, using standard TV transmission channels (terrestrial, satellite or cable). With almost 6 million digital TV subscribers in the UK, it is a matter of time and political will to turn off the analogue transmission for pure digital TV. The fact of analogue capture, transmission and display of moving pictures over the years produced a distinct TV industry fed by an ever increasing hunger for TV in all countries to reach practically most of the human population. [1]

Since 1970, around three decades after the start of TV for home reception, the Internet was introduced. Over the years, in particular by the introduction of the World Wide Web, the Internet has become the main medium for information exchange and communication, creating a universal space where everyone, with no restriction of geography or country. Any one can be connect and access the shared information and make its information accessible to all. Of course this has coincided also with the PC explosion bringing down the cost of a PC close to a TV set. Thus in many households PC started to take its place next to the TV and competing for viewing time. The telecommunication companies first using the existing infrastructure for voice transmission have started to develop infrastructure for digital data communications and the Internet. More recently, to respond to the needs of fast Internet connection, telecom companies started to introduce ADSL to take advantage of the copper wire that exists for telephony.

As TV technology converts to fully digital, the transmission medium so far used by broadcasters only as a "push" medium to transmit programmes becomes also a vast digital channel where all sorts of digital data including Internet and the web content can be transmitted. At the same time, the increased bandwidth to home provided by the telecomm operators to respond to the demands of the Web "pull" technology is also giving rise to a new medium for TV broadcasting called the Web Television. Of course, there is also the option to use this new medium for providing interactivity to the TV, combining the TV with commerce (t-commerce) and so on. We can also envisage including moving video images as an addition to the web pages. This paper will be studying some of the issues involved in combining TV and Internet. We will also mention a recent European project called SAMBITS, which is studying a novel way of combining the Internet and TV.

2. DIGITAL TELEVISION

Digital television (DTV) technology today involves digitally coding TV images compliant with the MPEG-2 standard and transmitting these digitally via a terrestrial antenna, cable or satellite. The received digital signal is connected to a device called "set-top box" which decodes the image and converts it to the analogue form, which can then be displayed by a standard TV set. Thus with such a hybrid receiver, we are able to receive conventional analogue TV channels as well as digital ones via the set top box. It is expected that the analogue channels one day will be turned off (in about 10 years time) and replaced by digital ones by which time all TV sets will have digital decoders built in thus integrating the set-top box with the TV set [2].

The driving force in Europe for digital TV has been the Digital Video Broadcasting (DVB) group formed in 1993 by the European manufacturers and broadcasters as an independent organisation to set standards [3]. DVB is based on standard resolution TV, 625 line, 50-Hz interlaced pictures with MPEG-2 standard video compression, and sound compression with a wide-screen variant of 16:9 aspect ratio. The optional high-definition mode doubles the number of lines to 1250 interlaced. DVB has so far produced three versions of the standard for satellite (DVB-S), cable (DVB-C) and terrestrial (DVB-T).

In the US the American Television Standards Committee (ATSC) operates under the aegis of the US Federal Communications Commission (FCC). ATSC is based on computer display standards with picture transmission rates of 24, 30, or 60 Hz to match cinema projection standards and the NTSC analogue TV system.

Figure 1 depicts the main components of a digital video broadcasting system using a satellite for transmission. The same model applies to other forms of transmission using cable or terrestrial mediums. In such a system, video streams

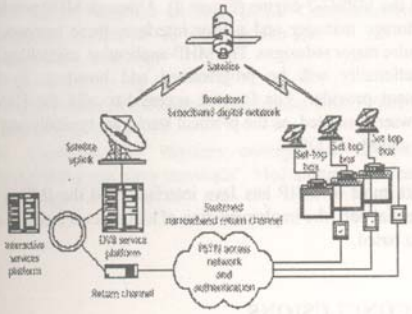


Figure.1. Components of digital video broadcasting system [1]

and audio is compressed according to the Moving Pictures Expert Group MPEG-2 standard algorithms [4]. A number of programmes are multiplexed for transport over a distribution network. This infrastructure provides a broadband digital distribution network, data transport protocols, and digital terminals (set-top boxes) at the user's premises. This in a sense is a powerful platform for delivery of digital data and information, either to enhance the regular TV programmes or provide distinct non-TV services. One main concern, however, is the direction of flow of data that is one-way and the same to all receivers (one-to-all). A return channel is often provided using the Public Switched Telephone Network (PSTN). This complicates the picture, however, since it involves another operator other than the broadcast service provider, not to mention that such a return channel is of low band with and introduces extra costs.

3. INTERNET OVER DVB

The MPEG-2 standard defines the way compressed video and audio are encoded as packetised elementary streams (PES) and transported. This is an asynchronous time division multiplexing system (TDM), which specifies the packet transmission of fixed length cell of 188 bytes. The packet header specifies the individual TV programmes and video, audio and control streams. System internal tables are transmitted in section packets. The data rates are selectable with 4 or 8 Mb/s as dictated by TV quality.

With such architecture there are three methods of carrying IP datagrams over MPEG-2 [5]:

1. Data packets can be encapsulated and carried inside the PES packets intended for TV,
2. Data packets can be carried inside the section packets intended for system internal tables,
3. An adaptation layer protocol is used to segment data packets directly into a sequence of cells.

MPEG-2 offers two service access points for data streaming. One uses PES packets and the other one table sections. In both cases a segmentation and re-assembly function has to be performed separately. The PES encapsulation is called *data streaming* and the method based on sections is called the *multiprotocol encapsulation (MPE)*. The third method is known as *data piping* which needs an explicit segmentation and reassembly layer. All the three methods involve a certain amount of overheads since the IP datagrams do not come in multiples of 184 bytes. In fact, the majority of IP datagrams for TCP, HTTP, or FTP packets are either 576 or 1500 bytes long. This causes an overhead of 13-15 per cent for MPE encapsulation [5].

4. MULTIMEDIA HOME PLATFORM (MHP)

Once the digital delivery of video to home became a practical prospect, a great deal_of effort was spent to define what should constitute a receiver. Clearly, such devices had to be standard, not tied to a particular network provider or manufacturer. Considering the scale of the market, and the variety of players, one can understand the level of interest. Such activities led to associations to develop the standards like DAVIC [6] and others. At first such efforts were directed towards the reception of digital TV, like the UNITEL-universal set-top box project launched by the European Comission in 1996. As later it was realised that the fact of transmitting digital signals also provided a channel for all sorts of digital services and in particular access to the Internet. About the same period of time a new platform independent application writing language Java has been widely used. Therefore it became clear that to be able to integrate other services to the digital TV and combine it in a seamless way with the Internet a more broad and universal approach was needed. This led the UNITEL initiative to become the MHP initiative for the DVB consortium [7].

The architecture of the MHP is defined in terms of resources, systems software and applications. For example MPEG processing, CPU, memory, and graphic system are MHP resources (Figure 2). The core of MHP is based around a platform known as DVB-J. This includes a virtual machine as defined in the Java Virtual Machine specification from Sun Microsystems.

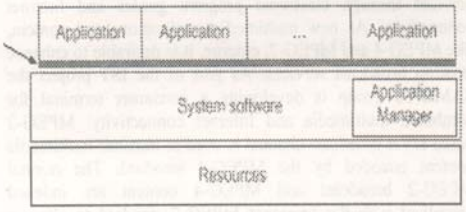


Figure 2. Basic MHP architecture

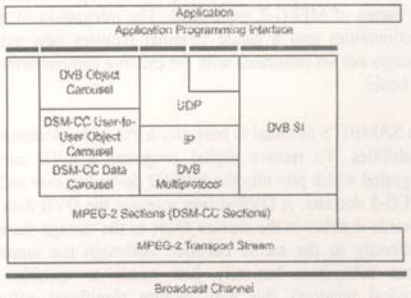


Figure 3. Broadcast Channel Protocol Stack

MHP defines a number of broadcast channel protocols as shown in Figure 3. Here we will mention the Data Carousel

concept, which provides a means of user selectivity with no return channel. The Data Carousel data stream consists of indexing and naming mechanism to locate objects which are repeatedly broadcast and received by a viewer when tuned to a channel. Figure 4 illustrates the set of DVB defined interaction channel protocols that are accessible by MHP applications.

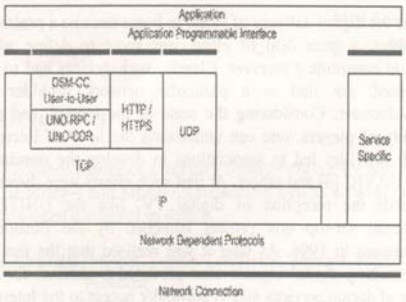


Figure 4. Interaction Channel Protocol Stack

5. SAMBITS PROJECT

The SAMBITS (Systems for Advanced Multimedia Broadcast and IT Services) project is supported by the European Union IST programme aimed at developing integrated and interactive broadcast services. A number of major European broadcasters, industries and academic institutions are partners to this international effort. The approach taken is consistent with the existing and emergent standards and set-top boxes. Nowadays set-top boxes are a common device for receiving and providing access to digital TV broadcast services. In addition customers get enhanced services through electronic program guides and Internet connectivity. As new multimedia applications and content, like MPEG-4 and MPEG-7, emerge, it is desirable to enhance existing broadcast services. As part of the IST project the SAMBITS group is developing a consumer terminal for combined multimedia and Internet connectivity. MPEG-2 based DVB broadcast channel is used to transmit multimedia content encoded by the MPEG-4 standard. The normal MPEG-2 broadcast and MPEG-4 content are indexed compliant with the emergent MPEG-7 standard to allow a user to interact with content, retrieve additional related information, and search for pre-specified sequence sections (by means of MPEG-7 meta data). The integration of these functionalities into a single terminal requires new system concepts not yet integrated with the existing conventional set-top boxes.

The SAMBITS terminal is basically a PC with additional IO capabilities. To receive digital programs a DVB card is integrated which provides the MPEG2 de-multiplexer and the MPEG-2 decoder. A DVB-driver accesses the DVB data and forwards it either to the *section filter*, to the storage manager or directly to the media decoder. Although the graphical output subsystem basically just needs to combine all graphical windows this may require significant software support, in particular with respect to 3D. Finally the graphical subsystem performs also the scan rate conversion to TV formats. On the software side the terminal is based on MHP, which provides all necessary functions of a set-top box. The major software modules to be added are the MPEG-4 player and the MPEG-7 engine (Figure 4). Although MHP provides a storage manager and a user interface, these components require major redesigns. The MHP-application controlling the functionality will be programmed and broadcast by the content provider. For Internet access basically the HTML browser is needed, as the protocol stacks are typically part of the underlying OS.

Since most of MHP has Java interfaces and the JMF is an essential part, the implementation of higher level modules are Java based.

6. CONCLUSIONS

As the set-top box technology and PC are converging thanks to the digital TV broadcasting and MHP, the increased bandwidth for the Internet services are providing a new channel as a carrier for TV. Thus it is clear that within a short span of time the home TV will look like a PC and a PC like a TV. The challenge lies, however, the opportunities that such a convergence provides in devising new services, which add interactivity and navigation capability to the existing digital, TV programmes, regardless the mode of delivery. In this respect the new standards for multimedia like MPEG-4 and MPEG-7 for meta-data will come to their own since they will provide means of interactive multimedia delivery and user agent assisted search and navigation. SAMBITS is one of the projects exploring such a scenario.

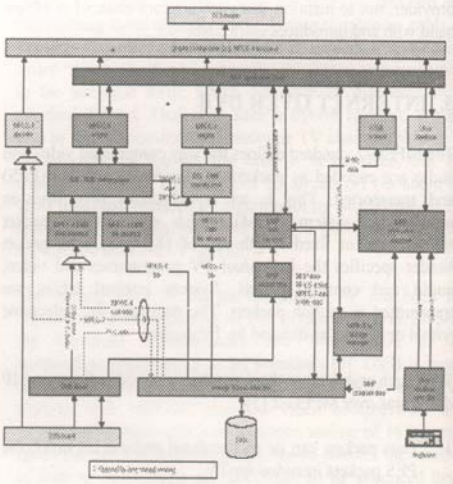


Figure 5. SAMBITS terminal

REFERENCES

[1] B Fox, "Digital TV comes down to earth", IEEE Spectrum, Vol. 35, No. 10, 1998, pp. 23-33

[2] M Milenkovic, "Delivering interactive services via a digital TV infratructure", IEEE Multimedia, Vol. 5, No. 4, 1998, pp. 34-43

[3] http://www.dvb.org

[4] http://www.mpeg.org

[5] H D Clausen, H Linder, and B Collini-Nocker, "Internet over direct broadcast satellites",IEEE Communicatrions Magazine, Vol 37, No. 6, 1999, pp. 146-151
[6] http://www.davic.org
[7] http://www.mhp.org
[8] Digital Video Broadcasting; Multimedia Home Platform (MHP) Specification 1.1
[9]K Illgner et. Al. "System concept for interactive broadcasting consumer terminals", Mediafutures, Florence, May 2001

Easy-to-use Multimedia Tools and Scalable Distributed Architectures for Web-based Teaching and Learning

Professor Chris R. Jesshope
Director NZEdSoft, Massey University, Private Bag 11222, Palmerston North, New Zealand
and
Department of Computer Science, The University of Hull
Hull, HU6 7RX, UK

ABSTRACT

Paper describes work being undertaken at both Hull University and Massey University in the development and deployment of tools for web-based teaching and learning. The tools described here comprise an easy-to-use multimedia authoring system and a scalable system for the management, authoring and delivery of education in both on-line and off-line modes. The former is the AudioGraph Recorder and is in use by a large number of teachers and trainers around the world. The paper will give a brief introduction to its capabilities, describe the work in progress and show how it has been used in practice. The educational delivery system is designed to be both flexible and scalable. To this end we have exploited both legacy and emerging standards in database technology, such as SQL and XML. The system is also designed to be scalable by utilising the network of students' computers as a distributed server. The architecture and implementation issues solved in developing this system will be discussed.

Keywords: Multimedia Authoring, Web-based Teaching and Learning, Web-based Education Management and Delivery.

1. WEB-BASED EDUCATION

Web-based education has been held up as a panacea for many years; our own work in this field started in 1995 and from very practical requirements. The initial reluctance to adopt is changing and we are now seeing a rapid take-up of this technology. A good example of one of the drivers is the UK initiative of the e-University [1]. This is typical of the widespread interest of governments and institutions in this use of the Web, together with the ubiquitous personal computers, to enable information to be exchanged more quickly and flexibly than ever before. People now expect information that is immediate, personal, current, engaging and collaborative [2].

Technology and knowledge are also the momentum of any new economy. There is always a shortage of skilled workers in the work force. Post-secondary or post tertiary education is now mandatory in the workforce due to the rapid and accelerating rate of change of many sciences and technologies. This requires a constant refreshing of knowledge in a skilled work force. This education will have to be provided by many new players and, in addition to the traditional schools and universities, new education providers, including the companies requiring the skilled workforce, will be brought into this new arena. This education may be provided locally, on a company's Intranet or may provided globally, with experts in a given niche selling their expertise to the world. The technology for this already exists, using any number of different delivery methods[3].

The list below includes only those relevant to web-based delivery, the most promising mode of delivery due to its widespread use.

- Compressed Video
- Audio/visual presentations
- Audio
- Electronic document exchange
- E-mail, Usenet
- Electronic classroom environments
- Internet or CD-ROM/DVD delivery

Web-based delivery of distance learning material can be provided synchronously (in real-time) or asynchronously (by recording and serving the material on a web site). Media such as audio, video, image, text and graphics can also be delivered in an interactive and above all a continuously monitored environment and the cost is affordable for all learners, regardless of their location.

Today's web browsers such as Netscape and Internet Explorer are so easy to use that even a computer illiterate person can grasp the basic skills in a very short time. These browsers are free to use and their functionality can be extended by either Java applets or by providing plug-in components.

This delivery mechanism is also dynamic, as it provides the learner with an interactive interface, integrating all kinds of media within a hypertext environment. "*The WWW is immediate, personal, current, engaging and collaborative. Learners are responsible for their own learning in a medium that allows exploration and discovery*"[2]. Web-base learning therefore helps the learner to develop skills such as how to collaborate and how to find knowledge from the extensive resources available. Validation of that knowledge is one of the major hurdles for the adoption of this direction in education. We will return to this issue later.

There are now significant political pressures to ensure the needs described above are met and education providers are pursuing technology to achieve these goals at an ever-increasing pace. It is predicted [4] that in the United States over the next two years, colleges and universities will use web-based learning as a means to improve education flexibility and to provide education opportunities to more students. With only 30 percent of classes now using web pages to distribute class materials and resources and less that 5 percent of campuses having adopted web-education delivery platforms, the adoption is still in its infancy. However, it is estimated that within the next two years, about 50 percents of all campuses will install a campus-wide learning platform.

2. DEVELOPMENT AND DELIVERY OF EDUCATIONAL MATERIAL

Searching the web reveals many thousands of websites that purport to deliver educational material. However, these so-called "web-based distance leaning systems" do little more than put lecture notes online or just create links to material. Although this is may be useful to augment a traditional education, it is not sufficient in itself to provide an on-line education and is certainly not exploiting the medium to its fullest. There is in fact a dearth of high-quality educational material. Even sites that contain animation, graphics, video or audio, tend only to use this approach for visual effect, without regard to pedagogical efficacy.

Concerning the delivery of this material, there are currently two approaches: to use a standard web server, with or without access restrictions, or to use one of the many educational web frameworks. These will include access checking, access logging, on-line communications, and perhaps some simple authoring e.g. for tests. Although most of these systems are very similar, there are numerous comparative evaluations to be found [5-8].

Given the extent of the material and tools available, why is there still a need for continuing development in this area? One of the reasons that most of the material available is text-based is due to the high cost of developing multimedia. This may require hundreds of hours preparation for every hour of presentation. This ratio is untenable and it is clear that the development of the media must be simplified in order to reduce the costs and at the same time bring the educators into the picture. AudioGraph [9-11], one of the tools described in this paper, has attempted to do just that. This tool will be described briefly in the next section. It has also been demonstrated that it can be used successfully to produce and deliver economical educational multimedia [12,13].

Returning to the web-based delivery of educational material, the existing tools[5-8] have been developed as ad-hoc manner extensions to a conventional web server. Most are based on server-side scripting, with Java providing for client side interactivity. None of these tools really provide standards for interfacing to legacy systems or for interoperability across other learning objects that can be found on the web. In our analysis we found the following deficiencies:

Lack of flexibility in the mode of delivery: all provide only online delivery. This can be both expensive and inflexible. All material has to be downloaded from the server rather than delivered on CD or DVD. In the case of video or even high-quality audio, this requires end-user bandwidths unavailable in most of the world. The problems that need to be solved in allowing an off-line access mode however, are not inconsiderable.

Lack of flexibility in interfacing with legacy systems: many current education providers already have large investment in database systems for administrating their business. Web-based systems must integrate seamlessly into these legacy systems, which will usually require SQL database interfaces.

Lack of adaptability: current tools are mostly designed for browsing and information retrieval, but not for active learning. Students with different background knowledge will essentially receive the same educational materials. Adaptivity in delivery is an important aspect of web-based learning systems. It is defined as the ability to be aware of user's behavior and knowledge, and to take this into account in providing the user with the right kind of learning object[14]. There is a conflict between this and off-line delivery requirement. On-line access can be monitored easily but in off-line mode, with material delivered on CD, special consideration has to be given to the logging of student activity and its use in adapting the presentations the student receives.

Lack of accuracy in multimedia document retrieval: A querying delivery system is normally achieved based on keyword searches. Accuracy of keyword searches is not good; normally this would yield either to few or too many matches. Semantic-based searches have the potential to provide much more accurate retrieval.

From this analysis, it is clear that there is also a need for further development of web-base distance learning systems to overcome these shortcomings. This is the origin of TILE project [15].

Both the AudioGraph and the TILE system will be described in this paper. Work on the AudioGraph is very mature with many users of this commercial-quality software. This project is described in section 3. The TILE project, on the other hand, is currently in the development stage. The design stage has been completed and the result of that is described here. We have already prototyped most of the system and will describe this in section 4.

3. THE AUDIOGRAPH TOOLS

AudioGraph was developed for on-line teaching and training. The tools comprise an authoring program called the AudioGraph Recorder and plug-ins for web browsers to play back the recorded presentations. The tools are available as a free download from http://www.nzedsoft.com/. Currently the Macintosh version is more mature, with version 1.3, the fourth release, having been made available in mid 2001. The Windows version, released at the same time, is a beta release (version 1.0beta). This section describes the tools and their planned further enhancement.

The fundamental philosophy in the design of the AudioGraph has been to assume that the tools would be used by academics, teachers and trainers, rather than by multimedia professionals, who are already catered for by a number sophisticated and complex authoring tools. This required a thorough analysis of the basic requirements of on-line education and the provision of an intuitive interface that allows these non-professionals to produce professional looking presentations very rapidly and without prior multimedia experience. This strategy places the development of multimedia firmly in the hands of the educators rather than the professionals. In this way, we hope to achieve a paradigm shift, in much the same way that paper-based publishing underwent with the advent of easy-to-use word processors.

At the outset of this project our goals were as follows:

- to target the delivery of multimedia course-ware via the world-wide web;
- to provide simple-to-use tools capable of producing multimedia presentations with little or no experience of multimedia editing;
- to significantly reduce the industry norm of 200 hours of preparation time for every hour of multimedia

presentation;
- to ensure the presentation size was small enough so that institutions could provide a large corpus of multimedia tuition on a modestly sized server;
- to ensure that the multimedia presentations could be accessed over modem connection speeds (e.g. 14K), without any significant delay;
- to provide cross platform delivery of the multi-media contents using a browser plug-in.

The fact that around 1000 users now use these tools worldwide is an indication that these objectives have been largely met. The tools are under constant review but we are resisting the temptation to add too many features to maintain ease of use.

An overview of the tools

We had to make compromises in the design of the tools. The major one is in the media supported. AudioGraph supports only compressed audio, images and vector graphics as its media elements, and generates presentations for web-based delivery. We use an iconic interface to and void synchronisation issues using the AudioGraph principle, which requires the presentation to be a strict sequence of media elements. Some of the media elements have real-time semantics, such as pauses and sounds. Others, such as the rendering of an image or vector graphic component, have no real time meaning but are rendered at the speed of the playback computer, which may be very different on different generations of computers. Whatever the playback speed however, the media file structure, which embeds the AudioGraph principle, maintains the strict sequence and hence the correct semantics of the presentation.

AudioGraph does not support video and clearly this is to satisfy the requirements of a small footprint website and low-bandwidth streaming media.

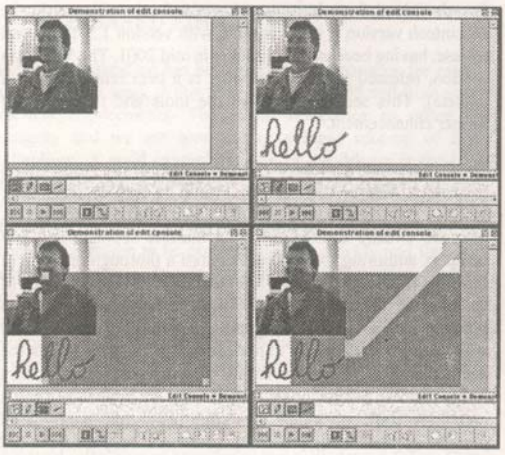


Figure 1. A composition of screen shots showing the edit console and slide window. This shows the correspondence between the iconic and graphical representation of each media element

Results from using the tools in extensive field trials are presented in [12,13], which demonstrate the effectiveness of the tools.

The editing interface

The main editing interface that implements the AudioGraph principle is a pair of windows: *the slide window*, which shows the presentation as it will appear at any point in the presentation sequence and the *edit console*, which represents the sequence of media elements in iconic form. The user can select any icon within the edit console and the slide window displays the presentation, as it would appear when that media element has been displayed. Figure 1 gives a demonstration of this principle. The four media elements in the edit console are an image, a freehand vector graphic element (the handwriting), a rectangle (notice the transparency) and a straight line, in that order. Notice that only those elements up to and including the icon selected in the edit console are displayed in the slide window. Any selected element can be moved in either window (in space or in time). The edit console shows the presentation as a sequence of media elements (iconically), which can be moved around on the temporal axis and the slide window shows a corresponding snapshot of the spatial representation of the presentation at any point on the temporal axis.

The edit console has a number of controls, which are for previewing the dynamic presentation. These navigate, play and stop playback. Other controls are used for editing and grouping together the media elements.

The tools

Figure 2 shows the tool menu, which is used to select a tool to place the media elements in the presentation. At the top of the menu, there are various vector graphic tools, namely (from left-to-right): freehand line - used for handwriting, straight lines, rectangle (open and filled), ellipses and finally, arcs. Then there is a highlighter, used like a highlighter pen and an eraser. Below these are the image placement tool and the link tool, then the scroll and the text tools and finally the sound and pause tools. The remainder of the window provides some tools for editing attributes and navigation. If a media element has been selected, then these tools modify its attributes, otherwise the attributes of the tools themselves is edited. In this window, colour and line thickness can be modified, which are the most commonly used attributes. Other attributes can be edited from a separate attributes menu.

The media elements– Images

A simple presentation style using the AudioGraph would typically use a set of presentation images, as might be produced

by PowerPoint for example, which are then annotated using audio, highlighting, handwriting graphics and possibly other images. This is not the only way that a presentation can be constructed and [12] gives examples of a number of different development techniques.

Images can also be imported from a screen capture, or cut or copied from another application. AudioGraph supports transparency in its vector graphics and when this is combined with the pixel-by-pixel transparency of the PNG graphics[17] used in the web presentations, some very sophisticated results can be achieved in a presentation.

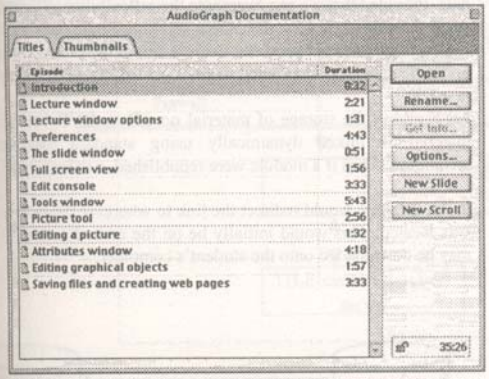


Figure 3. The Lecture window, showing the characteristic of each slide in a presentation, including duration of slides and total presentation.

Sounds and pauses –

The most data-intensive media element is sound. Fortunately there are very efficient compression algorithms for transmitting sound over a network. This is especially true if, as is expected, the sound input is going to speech. The AudioGraph uses the GSM compression technique [16], which is optimised for speech. An uncompressed speech stream of 16-bits accuracy and sampled at 8KHz would require 128 Kbits/sec for transmission over a network. When compressed using GSM, the 16-bit/sample speech would require only 13.2 Kbits/sec thus providing a compression ration of approximately 10 to 1. Additional compression is achieved by dividing the stream into sound bytes separated by explicit pauses of arbitrary duration. Sound can be recorded directly, using the sound tool and a microphone. Sound of any sample rate and precision supported by QuickTime can be cut or copied from other applications and pasted into the AudioGraph using the clipboard. Compression takes place, as with images, on generating a web site.

Websites

AudioGraph builds quite sophisticated web sites, without any web editing at all. The basic structure of the site comprises an index page of links, one for each of the presentation units (we call them slides or episodes) within the lecture document. Figure 3 shows the lecture interface, which lists all slides within a presentation. It allows the user to specify a meaningful name for each slide, lists the duration of each slide and the total duration of the lecture.

This window reflects the structure of the web site that will be generated for this presentation. Each slide will generate a link to it as a presentation. That presentation may be generated within the same window or may use a new window. The Options button allows various attributes of the slide to be set, such as size, background colour etc.

A presentation was generated from the lecture shown in figure 3. The index page produced is shown in figure 4 and it can be seen that it mirrors the lecture window structure and also includes

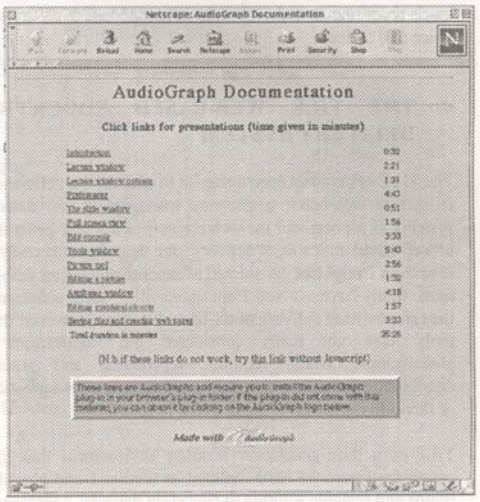


Figure 4. Web page index generated from the presentation given in figure 3.

instructions for playback, including the download of the plug-in.

Further developments

We are still developing the AudioGraph tools. Our current version produces media, which is sequential, like a movie. The only control available to the student is the ability to select the component of the presentation from within the html index page and to position the playback position from within the plug-in interface. Although this provides all of the navigation that is necessary, the effect is not as seamless as if the links were made from within the presentation itself rather than the embedding html page.

The next feature to be added will be the ability to activate objects within the media presentation as links, either to other AudioGraph presentations or to arbitrary URLs. This will allow users to generate adaptive presentations, where the presentation seen by an individual student will depend of the selections he or she has made. In effect the student will flatten a graph of nodes, each of which is an AudioGraph episode into their personalised presentation, based on the choices or answers they have given. In this way we will provide streaming presentations within which the students make committed choice.

This has been designed without compromising the already simple interface. Each episode will remain a strictly sequential presentation and the difference in interface will be small. A link attribute will be introduced at the slide window level and a new pane at the lecture window will summarise the link structure and allow the user to edit it.

Thus the change, which is quite a significant departure from the

current tools, will be affected with minimal change to the interface and yet at the same time will solve a major problem in multimedia streaming that provides scalability of streaming in the presence of choice.

Other features to be added will include compressed high-quality sound, the ability to add typed text as well as hand-written text and the use of volume activated detection of voice recording in order to reduce the bandwidth of captured voice still further.

4. THE TILE WEB-BASED EDUCATION DELIVERY SYSTEM

The TILE system has been designed to address the deficiencies in current web-based courseware management and delivery systems. If we consider the web browser as the first-generation of web-based educational delivery tool, then the tools analysed in section 2 might be considered as second-generation delivery tools. Many have evolved from universities' local solutions to the campus-wide delivery of on-line education. However, these tools have not really considered the implications of globalisation, which requires dependable and scalable distributed systems. Neither do these solutions offer a great deal of flexibility in terms of the delivery to the student.

TILE is a third-generation learning environment that must match the future we have outlined in section 1 and provide a robust and scalable solution to education delivery. Moreover, it must also meet the flexibility requirements of this new generation of learners, who will not be confined to our university campuses. TILE is a large collaborative project[15].

Let us consider the student's requirements first. The student may wish to study in a number of different situations:

- at home using their home computers - we assume that they have on-line access but that there is a finite cost for that access (it may be a time-based cost or, in the future, a packet-based cost where they are on-line continuously). Either way, we assume there is an advantage to reducing either cost;
- while travelling, with a laptop or similar portable computer - in this situation the student may have no access whatsoever to an on-line connection, or if they do it may be very expensive;
- at a conference, an Internet café or in a laboratory, using a computer, which they do not own and may not install software on..

This requires the support for both on-line and off-line modes of delivery, which in itself is not a difficult problem; it only becomes a problem when we also consider the requirements of the educator:

- the lecturer will probably publish the course once per academic period but may also wish to publish an update of it while the students are studying. If the material had been delivered on CD ROM for off-line access this would cause of problem of version control;
- the lecturer may provide adaptive material, in which case some kind of access logging must be maintained locally and this again is incompatible with off-line CD ROM access.

We have been searching for a single integrated solution that captures all of these conflicting requirements.

Another issue is scalability, we must be able to distribute data intensive material to the student cheaply. If we use on-line data delivery, scalability means providing faster and faster servers and eventually creating networks of servers, with their associated cost. The computer scientist E.E. Dykstra once said of computer networks "never underestimate the bandwidth of a truck-load of tapes". Well, today it would be CD and tomorrow DVD but certainly we should not underestimate conventional distribution channels, such as posting material on CD or DVD to the student; this is eminently scalable. The characteristics of this approach are a high bandwidth with a relatively high delivery time (days instead of seconds) and of course, static data. Provided that we can overcome the inflexibility of having static data, this is a very acceptable solution. Static data is not such a large problem provided that we can allow producers to update the material distributed without publishing a completely new CD.

TILE allows the storage of material on CD or DVD but this material is linked dynamically using standard database techniques. Thus if a module were republished, version control in the database would redirect the link to wherever the module was located. This would initially be on the central server but may be downloaded onto the student's computer.

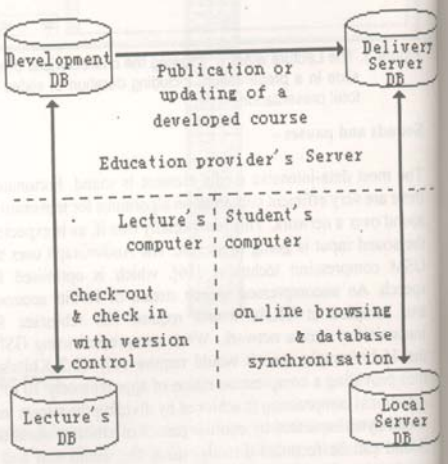


Figure 5. Courseware development and delivery cycle and the databases involved.

The second issue with off-line access is adaptive delivery. For pedagogical reasons, we may wish to monitor the student's progress in learning that material, building up models of their knowledge and preferred methods of learning [18,19] and then use these models to adapt the delivery of material on a student-by-student basis. This would not normally be possible unless the student was on-line. TILE allows adaptation in both on- and off-line modes by monitoring and logging students' choices and answers at all times. This also requires distributing the database that manages the course structure as well as the models held about a given student. Figure 5 gives an illustration of the cycle of courseware development and the databases, which must be maintained and synchronized in order to implement the above solutions.

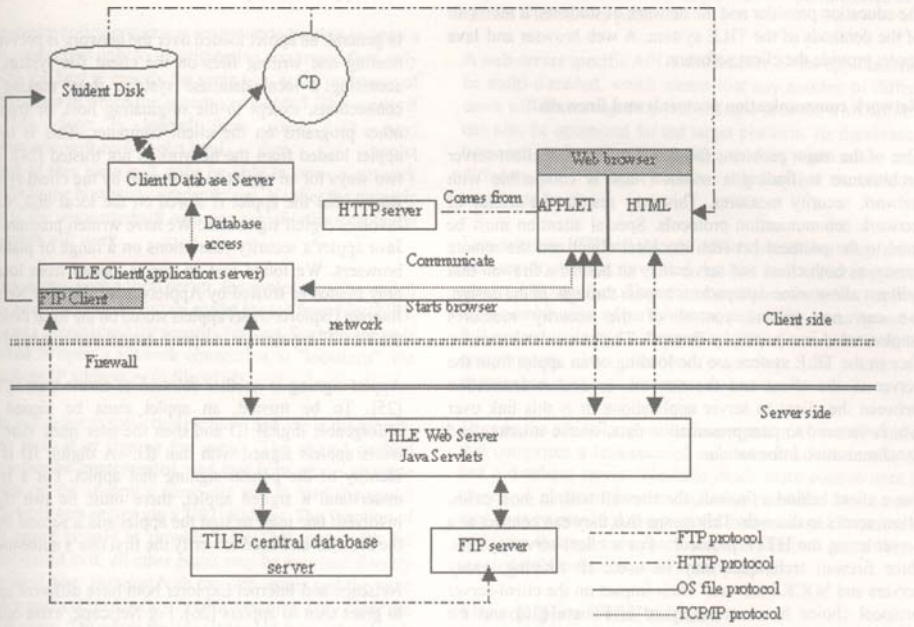


Figure 6. The TILE system architecture

In this paper we will look in detail at the delivery system (RHS of Figure 5). This provides a number of different modes of access for the student to the educational material:

- *On-line with local server* - in this mode, most of the educational material is provided from the student's local computer having been distributed on CD or DVD. Additional software (the local server) is required on the student's computer to manage and monitor its delivery. The student still has access to the on-line archive on the education provider's remote server(s) and local information can be updated from this source.
- *Off-line with local server* - in this mode the student has no access to on-line material, but the local server software continues to manage and monitor the student's progress.
- *On-line with no local server* - in this mode, the student for whatever reason is unable to install the local software and must access all material from the education provider's remote server(s).

In the latter two cases, inconsistency may arise between the information on the student's computer and that on the education provider's remote server(s). This may be due to the publication of new material on the remote server or updates in the student's logs or models. In either case, TILE implements mechanisms to synchronise inconsistent information between server and student databases.

The tile architecture

A client-server approach –

The architecture of the TILE delivery system is a client/server one. We use a thin client to present the information to the student and the server keeps track of and structures the information to be delivered to the student. The lecturer or content provider will define the structure, using an authoring interface tool (LHS of Figure 5). This may be modified by the students' models or indeed, may be overridden by the students themselves. In order to provide such flexibility, the system is database driven, using SQL for the students' and education providers' servers. Data exchange between the databases and the database on the lecturers' computers use an XML representation of the structure. The flexibility provided by the database allows for the distribution, updating and synchronisation of material.

Different modes of access have forced us to modify the traditional client-server architecture. In effect we distribute the server function to the students' computers. This provides the ability to monitor and adapt in the off-line mode and also aids the scalability of the system. Figure 6 illustrates this. It can be seen that a significant amount of the server functionality has been replicated on the student's computer, including a web server to provide the thin client to overcome security restrictions. There is a potential problem here as we have no control over the student's environment. Thus the local software must work with any operating system, web browser, etc. In short the local server must be completely cross-platform compatible. The minimum we can impose on the students' system is a Java-enabled web-browser, in which the client user-interface is implemented.

The result is a distributed three-tier, client-server architecture. The user interface, course-delivery logic and data access are separated and the client is now only concerned with the user-interface and connection to the server. Most of the course-delivery logic is moved into the middle layer, where it communicates directly with the database. This thin client architecture provides increased security, flexible and

significantly easier maintenance [20]. Our addition to this standard approach is to distribute the server function between the education provider and the network of students. It meets all of the demands of the TILE system. A web browser and Java applet provide the client software.

Network communication protocols and firewalls

One of the major problems faced in the design of client-server architecture is finding a solution that is compatible with network security measures. This may restrict the choice of network communication protocols. Special attention must be paid to the protocol between the local client and the remote server, as both client and server may sit behind a firewall that will not allow some data packets to pass through. In the design, we can not assume control of the security measures implemented in a particular firewall. The particular issues we face in the TILE system are the loading of an applet from the server to the client and the network or socket connection between the client to server applications. It is this link over which we need to pass presentation data, course structure and synchronisation information.

For a client behind a firewall, the firewall will, in most cases, allow access to the web. This means that they can connect to a server using the HTTP protocol. For a client-server system, three firewall techniques may be used: IP filtering, proxy servers and SOCKS servers. Their impact on the client-server protocol choice has been analysed elsewhere [21] and the conclusion drawn is that for complete generality, the HTTP protocol is the best choice of protocol.

Cross-platform issues

As already mentioned, the software on the student's computer must be platform independent. We have the choice therefore, to write the application for each platform supported or alternatively to use Java because of its ability to run anywhere. This provides a powerful reason for developing all software for the student's computer in Java.

For the client, web pages and Java applets provide the logic that is required for the adaptive presentation of the educational material. We have to provide the student with a local server and database, to support the other two TILE access modes: *online with local server and offline with local server* modes. Both server and database must be able to run on any platform. We have also developed the server software in Java. Although there are many database systems available, few meet the dual requirements of cross-platform support and low cost. A native Java database would meet our requirements provided that performance is not an issue. There are several open-source Java database systems available [e.g.22,23]. Performance should not be an issue, as the database will only be supporting a single user, the student.

Although it would seem that this strategy solves all of the cross-platform problems, this is not the case. There are several versions of Java development kits (called JDKs by Sun Microsystems). Macintosh platforms will only support JDK1.1.8 on Mac OS 9 and below. JDK1.2, the current standard, is supported on most other systems, but will only be available from Mac OS X. It is imperative that we support the lowest common denominator in developing the TILE system. We are installing software on the student's computer and must not force the student to upgrade either computer or operating system. Adopting JDK1.1.8 is the only answer.

Java applet security issues

In general, an applet loaded over the network is prevented from reading and writing files on the client file system, or from accessing a local database system, from making network connections, except to the originating host, or from starting other programs on the client computer. This is because an applet loaded from the network is not trusted [24]. There are two ways for an applet to be trusted by the client system. One way is that the applet is stored on the local disk, the second involves digital signatures. We have written programs to test a Java applet's security restrictions on a range of platforms and browsers. We found that an applet loaded from local disk is only treated as trusted by Appletviewer. Neither Netscape nor Internet Explorer trusts applets stored on the local disk to access the local file system or to open a connection to a local database.

Applet signing is another solution to create trust in an applet [25]. To be trusted, an applet must be signed with an unforgeable digital ID and then the user must state that s/he trusts applets signed with that ID. A digital ID is proof of identity of the person signing that applet. For a browser to understand a signed applet, there must be two digital IDs involved: one used to sign the applet and a second installed in the browser and used to verify the first one's authenticity.

Netscape and Internet Explorer both have different approaches to grant trust to applets [26]. For Netscape, extra code, which uses Netscape's specific Java classes, must be added into the applet. Signing methods are also different. For Netscape, the files must be signed with a Netscape Object Signing ID (creating a "manifest" of the files), and then the files and manifest must be wrapped into a .jar archive. For Explorer, the files must be wrapped into a .cab archive and then signed with a Microsoft Authenticode ID. Authenticode signing and Netscape Object Signing are not supported by all platforms. The ways that Netscape and Internet Explorer deal with expired digital IDs is also different. However, the most important and fatal issue concerning the use of signed applets to ensure security is that not all versions of Netscape and Internet Explorer understand signed applets.

Student-side application

Since neither a Java applet loaded from local disk nor a signed applet can provide a satisfactory solution to the requirements of accessing local files and connecting to the database system, another solution must be found. Because there are no security restrictions on Java applications for JDK1.1, the Java local server application and database can provide all local functionality, providing that we can establish a connection between this Java server and the Java applet. When there is a need to access the local file system, the applet sends a request to the server application and it accesses the database or file system. However, because both applet and application run on different Java Virtual Machines (JVM), there is no way for them to share memory. Again, a client-server approach is the only solution. The Java applet is the client, which opens network connections to the local Java server application, which accepts network requests from that applet. The user interface logic can therefore be separated from the courseware delivery logic, with the user interface provided by the Java applet and all delivery logic by the local Java server application. However, there is still a problem in communication between the Java applet and the local Java server application.

Communication between the Java applet and the local server

Java's security policy constrains an applet to be able to open a network connection only to its originating server and then only by naming the host in exactly the same way as the hostname of the HTML page in which the applet is embedded. This means if the HTML page is loaded from some URL, say: http://www.nzedsoft.com/applet.html, then the applet embedded in this page will only be able to open a network connection to the host www.nzedsoft.com. Neither the numeric IP address for www.nzedsoft.com nor any shorthand notation for that name will work.

For offline access with a local server, the applet, must be loaded from the local file system. But an applet loaded from file://URL is treated as a special case for security reasons. Such an applet is not allowed to open a network connection to "localhost", the local machine IP address or to file://URL.

There are two ways to solve this problem and one is the already discarded method of signing the applet. The second method must therefore be implemented, and that is to set up a simple web server on the student's computer and to load the applet from this local web server via a URL address. The function of this web server is only to provide the HTML page with the applet embedded in it, all other pages may be accessed directly from the local disc. Because both the web server and the local Java server application run on the same machine, they share the same host name. Therefore the applet loaded from a local web server may open a network connection to the local Java server application.

Thus, on the student's computer we must install a system, which comprises a database, a limited web server and the local Java server application. The applet that provides the user interface will be loaded from this local system, when the student is using their own computer, or from the education provider's server, when the student is using an anonymous computer. These components have all been implemented in Java JDK 1.1.8.

Software required on the education provider's computers

On the education provider's computers, for firewall security reasons, all transactions with the student must use the HTTP protocol. The requirements of the education provider's server have therefore been met by adding functionality to a standard web server. In this situation however, performance is very much an issue. Although we have offloaded many of the transactions onto the student's local computer, any remaining transactions, to do with synchronisation and anonymous computer use, must be implemented as efficiently as possible to ensure good scalability. This server-side logic may be implemented using a number of different techniques:

- Common Gateway Interface (CGI);
- Web-server specific Application Programming Interface (API); or
- Java Servlets.

CGI is a very flexible technique and is used in many small-scale applications. It can be used to develop an application on any web server and on any platform, and it is programming language independent. Its disadvantages are bad performance and poor scalability, coupled with potential security holes. The scalability is poor, as each new user request requires a new process to be created on the server computer, and if the number of processes created is very large, then the performance of the system will be very poor.

A web-server specific API is much faster, as the application can be multi-threaded, which means that any number of different users will be managed by just a single process. This approach can also be optimized for the target platform. Its disadvantage however, is its poor portability, as each web server has a different API. For legacy reasons this may not therefore be a sound approach.

Java servlets [27] are also an API approach but they are not web-server specific, provided that the web server supports a servlet API interface. Thus the advantages of a servlet approach are a combination of performance, portability, security and the full access to Java's functionality [26]. From our feasibility comparisons, Java servlets outperform the other two approaches.

Thus on the education provider's computers we install a system that comprises a Java-enabled web server, a Java servlet API and a database server. We have much more control over the choice of platform for the education provider; the computer may even be provided as a part of a total package. Therefore the problems that we have faced in cross-platform portability on the student's computers will not apply.

The complete TILE solution

The overall architecture of the TILE education delivery system is shown in Figure 6. It is an extension to the classical three-tier, client-server architecture. Students interact with the system using a standard web browser. The intelligent user interfaces are provided by the Java applet. For offline with local server mode, the delivery logic is provided by the local TILE server application and a local database server provides the data services. For online with no local server mode, the delivery logic is provided in the remote TILE-enhanced Web Server and a remote Central Database Server provides the data services. For online with local server mode, the delivery logic and data services may be provided from either local or remote system, as appropriate.

5. CONCLUSIONS

We have shown that despite the profusion of tools available for web-based education, there are still problems that must be solved. We have looked at two broad issues, the development of web-based multimedia, which, we believe, if it is to be taken up seriously must be usable by non-experts. We have described tools that are available and, which are still being developed that solve this issue. The second issue is in providing both scalability and flexibility in web-based delivery. This has been solved using distributed systems techniques.

In looking at this latter issue we have analysed the requirements for a third-generation, technology-integrated learning environment from the perspective of both performance and security as related to the implementation of a general distributed system. We have also analysed the student's requirements in terms of modes of access and flexibility of delivery, which will support a range of good pedagogical practices. Our final constraint was to produce a system that did not force the learner into updating his or her computer to a

particular operating system or specification.

The conclusions that we have arrived at are that systems design constraints as well as pedagogical requirements can be satisfied by the architecture that we have proposed in this paper. This architecture does seem complex but the implementation may make use of many existing components. Moreover issues such as installation can be automated and software updates need only be made available to the server.

Data, such as the structure of a course, students' models etc. will be abstracted using SQL databases. This and the delivery logic is stored in one or both of the servers, the one on the education provider's side and/or the one on the student's side. A prototype of this system has been implemented using Linux, Apache and Tomcat on the server side and using a Java applet, server, database and web server on the student's side. Work is continuing with the development of authoring interface tools to develop the course structure within the database and link to various media authoring tools such as AudioGraph.

6. ACKNOWLEDGEMENTS

We would like to acknowledge the support for this project from the New Zealand government's New Economy Research Fund (NERF) under contract MAUX9911.

REFERENCES

[1] Paul Maharg (2001) The e-University Project: what's next after UNext?, http://www.ukcle.ac.uk/news/directions9.html(retrieved on 13/6/0).

[2] Wayne C. Poncia etc. (1999), Web-based Learning for a Digital Generation, http://www.nlginc.com/nlglibrary/articles/webbasedlearning.html (retrieved on 8/2/01).

[3] Barbara E. Truman (1995), Distance Education in Post Secondary Institutions and Business,University of Central Florida, http://pegasus.cc.ucf.edu/~btruman/dist-lr.html (retrieved on 8/2/2001)

[4] Wayne C. Poncia etc. (1999), Web-based Learning for a Digital Generation, http://www.nlginc.com/nlglibrary/articles/webbasedlearning.html. (retrieved on. 8/2/2001)

[5] Features/Tools and Tech Info, Comparison Table for all applications (Jan. 3,2001), http://www.ctt.bc.ca/landonline/choices.html (retrieved on 8/2/2001)

[6] Web-Based Educational Environments (1998), http://isis.acomp.usf.edu/Web-Based/support.html (retrieved on 8/2/2001)

[7] Tools for Developing Interactive Academic Web Courses, http://www.umanitoba.ca/ip/tools/courseware/evalmain.html (retrieved on 8/2/2001)

[8] Overview of Available Web-based Course Management Systems, http://www.cs.uml.edu/~heines/gowri/cmslist.html (retrieved on 8/2/2001)

[9] Jesshope, C.R. and Shafarenko, A. (1997) Web Based Teaching: a minimalist approach, Proc. Second Australasian Conference on Computer Science Education, ISBN: 0-89791-958-0, pp16-23.

[10] Jesshope, C., Shafarenko, A and Slusanschi, H. (1998) Low-bandwidth multimedia tools for web-based lecture publishing, IEE Engineering Science and Educational Journal, 7 (4), pp148-154, also published in IEE Computing and Control Engineering Journal, 9 (4), pp156-162 and online at IEE Computing Forum, http://forum.iee.org.uk/forum/library/, September 1989.

[11] R. Gehne and C. R. Jesshope (2000) Tools for the production of small-footprint, low-bandwidth, streaming multi-media for distance education, Proc Lifelong Learning Conference, Central University of Queensland (Brisbane, Australia), ISBN 187 6674 06 7, pp240-244.

[12] C. R. Jesshope (2000) The use of multi-media in internal and extramural teaching, Proc Lifelong Learning Conference, Central University of Queensland (Brisbane, Australia), ISBN 187 6674 06 7. pp257-262.

[13] C. R. Jesshope (2000) Using AudioGraph in On-line Teaching, Proc Open Learning Conference, Brisbane, Australia, pp315-320, Learning Network Queensland (Brisbane ,Austraalia).

[14] Barra, M.., Negro, A. and Scarano, V. (1999), When the teacher learns: A Model for Symmetric Adaptivity. In: Brusilovsky, P. and De Bra. P. (eds.) Proceedings of Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, Banff, Canada.

[15] C. R. Jesshope, Integrated tools for on-line education, Proc IWALT 2000, Massey University, New Zealand, IEEE Computer Society (Los Alamitos CA, USA), 2000,205-208.

[16] Scourias, J. (1999) Overview of the Global System for Mobile Communications, Retrieved from the web on 1/2/00, http://ccnga.uwaterloo.ca/~jscouria/GSM/gsmreport.html

[17] Roelofs, G (2000) Portable Network Graphics, Retrieved from the web on 1/2/00, http://www.cdrom.com/pub/png/

[18] Nikov A. & Pohl W. ,Combining User and, User Modelling for User-Adaptivity Systems. Human Computer Interaction - Ergonomics and User Interfaces ,1999 (Eds. H.-J. Bullinger & J. Ziegler).

[19] [12] Hoschka P. ,Computers as Assistants: A New Generation of Support Systems.(Lawrence Erlbaum Associates Publishers, Mahwah, NJ, New Jersey, 1996) 336-340 (ISBN 0-8058-3391-9).

[20] Joseph T. Sinclair and Mark Merkow, Thin Clients Clearly explained (Morgan Kaufman, Academic Press,2000).

[21] Marco Pistoia et. al. Java 2 Network Security, second edition (Upper Saddle River, N.J. ; London : Prentice Hall,1999).

[22] InstantDB, InstantDB home page, http://instantdb.enhydra.org (retrieved 6/6/2001)

[23] Hypersonic SQL, http://hsqldb.sourceforge.net/ (retrieved on 6/6/ 2001)

[24] Sun Microsystems (retrieved 6/6/2001) Frequently Asked Questions - Java Security, http://java.sun.com/sfaq/

[25] Daniel Griscom Code Signing for Java Applets, http://www.suitable.com/CodeSigningOverview.shtml#top (retrieved 6/6/2001)

[26] Robin Green Overcoming Java Security Problems on Netscape and IE, http://redrival.com/greenrd/javasec.html (retrieved 6/62001)

[27] Sun Microsystems Overview of Servlets, http://web2.java.sun.com/docs/books/tutorial/servlets/overview/index.html (retrieved 6/6/2001)

Parallel Processing for Fractal Image Compression Based on Full Searching and Hexagonal Partitioning

Ghim-Hwee Ong And Lixin Fan
School of Computing, National University of Singapore
Singapore, 117543
Email: {onggh, fanlx}@comp.nus.edu.sg

ABSTRACT

This paper presents a hexagonal partitioning method for fractal image compression. The method makes use of self-similarities existing in an image and applies the concept of hexagonal clusters to generate image blocks at different resolutions under a set of affine transformations in order to achieve better image compression and fidelity. Experimental results show that the reconstruction quality for detail-dominant images using hexagonal partitioning is better than those of other partitioning schemes such as quadtree partitioning. Also, the full search hexagonal-based encoding algorithm is suitable for parallelization due to its relatively heavy computational load. By running the parallelized encoding algorithm on multiple processors, the encoding time is drastically reduced while the reconstruction quality is retained. A speedup of about 10 can be obtained by using 13 processors.

Keywords: Image Compression, Fractals, Parallel Processing.

1. INTRODUCTION

Many image compression techniques, including both lossless and lossy, have been developed to suit different applications [1, 2]. Fractal image compression is one of the lossy data compression techniques that have been developed in the last decade [3]. It makes use of self-similarities existing in an image at different resolutions under a set of affine transformations and exploits this kind of redundancy in order to achieve compression.

An important issue involved in fractal image compression is the partitioning of an image into blocks for encoding. One such commonly used scheme is based on quadtree partitioning [4]. In this paper, we present a new approach to fractal image compression with hexagonal partitioning. However, the encoding of hexagonal-based fractal compression using a full search scheme is rather computational intensive. We propose a parallel processing algorithm for hexagonal-based encoding to reduce the processing time.

2. FRACTAL IMAGE COMPRESSION

In the last decade, fractal theory [3] has been applied to image compression. This is due to the recognition that fractals can describe natural scenes better than shapes of traditional geometry. In general, fractal image compression relies on the fact that all objects contain redundant information in the form of similar, repeating patterns called fractals. Objects in an image are then encoded as a set of mathematical data that describes the fractal properties of the image.

Based on different characteristics of fractals, many fractal image compression methods have been developed [4, 5]. Most of the fractal based methods were inspired by Barnsley's collage theorem [3]. In general, it is easy to view the space of all gray-scale or color images structured as a complete metric space with, for example, the Euclidian distance. However, it is in practice unlikely to find collages for most real images in terms of contracted copies of themselves by reasonably simple transforms. The practical approach to fractal image compression was proposed and developed by Jacquin [4] and Fisher [5]. Instead of trying to solve the inverse problem by matching parts of an image with the entire image by contractive transforms, the image itself is split into two categories of blocks: range blocks and domain blocks. That is, the image f is divided into range blocks $R_1, R_2, \ldots, R_i, \ldots, R_n$, so that $f = R_1 \cup R_2 \cup \ldots \cup Ri \cup \ldots \cup R_n$, and $Ri \cap R_j = 0$, where $i \neq j$. The range blocks cover the entire image and they do not overlap. The image is also divided into domain blocks $D_1, D_2, \ldots, D_i, \ldots, D_m$. Domain blocks can overlap and are larger than range blocks. For each range block R_i, we search for a matching domain block D_j among all the domain blocks so that the transformed D_j with a contractive transform w_i is similar to R_i, i.e., $R_i \approx w_i(D_j)$. Because of the image partition into range and domain blocks, the set of transformations is called a partitioned iterated functions system (PIFS). That is, PIFS = $\{w_1, w_2, \ldots, w_n\}$.

3. HEXAGONAL PARTITIONING

The proposed partitioning scheme for both the range and domain blocks in this project is based on the concept of *hexagonal cluster* and the operations on these clusters [6]. A cluster with size indicator k can be divided into seven smaller clusters with size indicator -1; on the other hand, seven adjacent clusters with size indicator k can merge into a larger cluster with size indicator k+1. This concept is actually the underlying idea of *hexagonal partitioning* which will be described as follows.

Let us first define the *array of clusters*. An *array of clusters* is a group of hexagonal clusters, which is of size indicator k and is placed adjacent to each other in a non-overlapping manner forming a region of a particular shape. This *array of clusters* or *array* can then be used to cover square image.

As depicted in Figure 1, the array can roughly cover the square image. Since each hexagon in the cluster is a *pixel* and each cluster is a *block*, the entire image is then divided into a number of blocks. These blocks are called the initial *range blocks*. According to our definition, each initial block is actually a cluster with size indicator k. For example, in Figure 1, there are 21 initial range blocks with size indicator 2. It is worth mentioning that the array of cluster in Figure 1 is not

drawn to scale for explanation purposes. In practice, the hexagon is much smaller and there are much more clusters to cover the entire image of normal size, say 512×512 pixels.

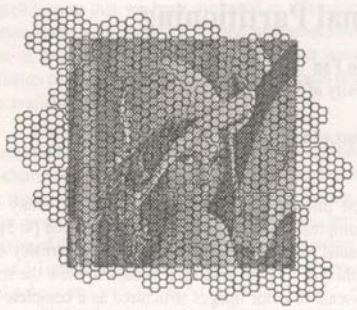


Figure 1. Array of hexagonal clusters.

Each initial range block can be partitioned into seven smaller sub-blocks, i.e., smaller clusters with size indicator -1. The sub-blocks can be further partitioned into even smaller sub-blocks, and so on. Therefore, the entire image might be partitioned into a number of initial range blocks and/or their sub-blocks depending on the image-contents and partitioning parameters. We call such a partitioning scheme the hexagonal cluster based partitioning or *hexagonal partitioning*. One possible partitioning result is illustrated in Figure 2.

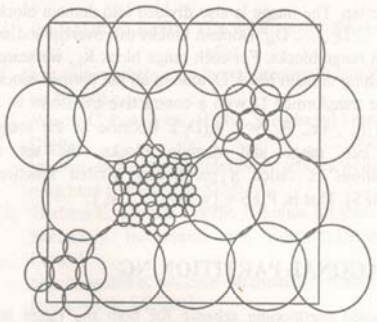


Figure 2. Hexagonal partitioning on a square image.

It can be seen that the hexagonal partitioning has the following properties: (1) The range blocks and sub-blocks have complex shapes which are dependent on the clusters size. The larger the size, the more complex is the shape. (2) Each block, if necessary, can be divided into seven sub-blocks. Compared with QuadTree partitioning scheme, it shows a tendency to partition given blocks into more smaller blocks, especially at regions containing details.

One highlight is that we encounter boundary mismatches when using an array of clusters to cover a square image. This is due to the fact that hexagonal clusters have jagged boundaries and it is impossible for these clusters to absolutely fit a square with straight boundary. Therefore, some hexagons in clusters do not correspond to any parts of the image. In our research, we assign these hexagons with predetermined grey intensities when we encode the clusters.

In practice, a real device, which can generate hexagonal pixels, is very difficult to find. Therefore, the implementation of hexagonal partitioning scheme in our research is still based on the traditional square grid. Figure 3 demonstrates how to construct *hexagonal clusters* on a square grid. This is achieved by means of mapping a square pixel to appropriate locations to form a *hexagonal cluster* H_1.

4. ENCODING ALGORITHM

Figure 4 depicts the sequential width-first encoding algorithm applying the full search scheme for hexagonal-based fractal image compression. Function **Compare** takes a range R and a domain D as input, computes and outputs rms, s and o. (The rms denotes the *root mean square* metric used to measure distance between two blocks. The s and o respectively denote the *contrast scaling factor* and *brightness offset* of transformation applied on domain blocks. The t denotes the predetermined threshold for maximum distance between range and acceptable domain blocks.) The **EncodeBlocks** function does not divide the unmatched range blocks into smaller sub-blocks. Instead, it marks each unmatched block as ***unmatched*** only. After all range blocks of same size have been examined and marked either ***matched*** or ***unmatched***, the **RangeDecomposition** function starts to check all blocks at this level. The blocks with mark ***unmatched*** will be partitioned into smaller sub-blocks and be sent back to step 3 for further encoding, while those matched blocks will be stored. Note that at next iteration, the current size is subtracted by 1.

In the algorithm, neither **EncodeBlocks** nor **RangeDecomposition** functions are recursive functions. The encoding time of a given range block at a particular iteration equals to (1) the execution time of comparing range block with each domain block multiplied by (2) the number of domain blocks in the pool. Since both (1) and (2) are constant at the particular iteration, therefore, the encoding time of all range blocks at a particular iteration are fixed and can be predicted beforehand. This uniqueness of encoding time makes the encoding algorithm suitable for parallelization.

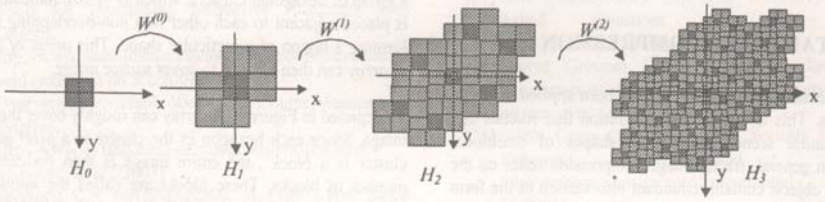


Figure 3. Hexagonal cluster on square grid (right shape).

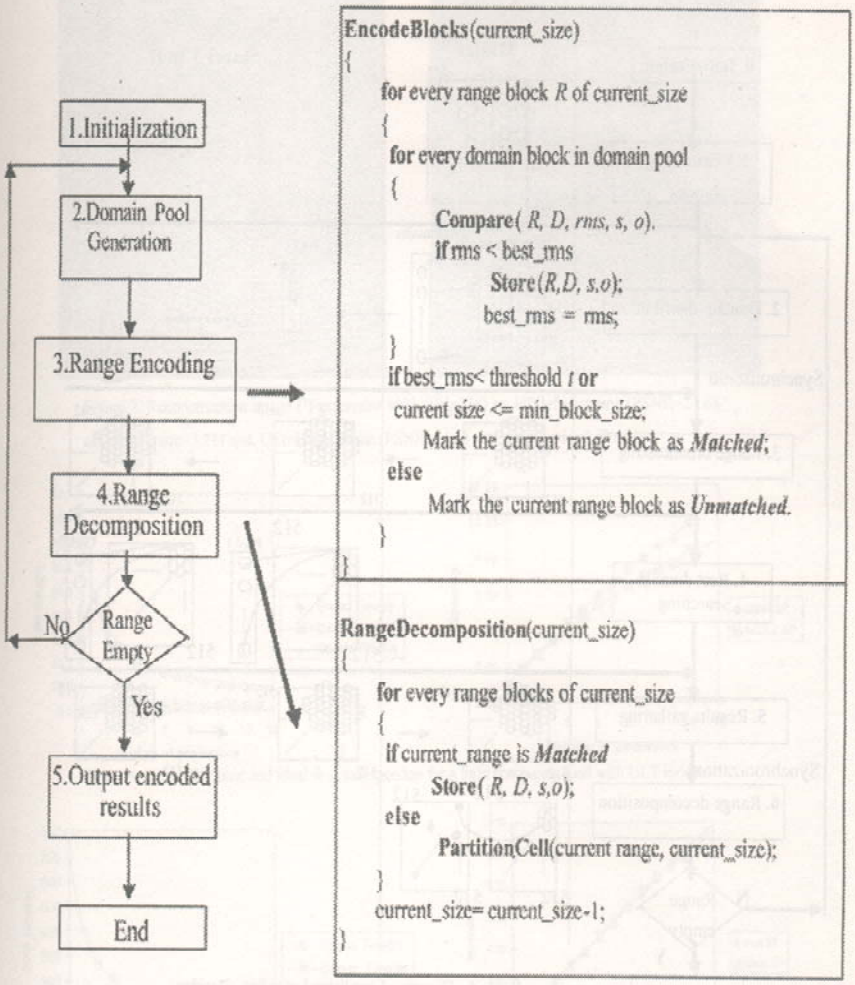


Figure 4. Diagram of the sequential width-first encoding algorithm.

5. PARALLELIZING THE SEQUENCIAL ALGORITHM

We now illustrate how to develop parallel algorithms based on the width-first sequential algorithm. Note that there are nested loops in the function **EncodeBlocks** (Figure 4). This large amount of loop iterations suggests a parallelizing speed-up potential. In the context of parallel programming, loops are considered as one of the largest sources of parallelism. Loops can be parallelized by assigning different loop iterations to different processors. The key question is how to properly schedule loop iterations in order to improve overall performance. For the function **EncodeBlocks** with a given current_size, we can either assign the inner loop or the outer loop to different processors. In the former case, domain blocks in the domain pool are distributed to various processors and thus searching for the best domain is limited by the corresponding subsets of domain pool on each processor. The latter method distributes all range blocks of a particular size to different processors and each processor will only encode blocks assigned to it. These two approaches are called *domain distribution* and *range distribution* respectively.

Another important issue to be decided is the scheduling scheme. Sine the encoding time of range blocks are fixed and can be predicted beforehand, the static scheduling scheme is suitable for this case. Therefore, there are two scheduling schemes available, namely, *static domain distribution* (SDD) and *static range distribution* (SRD). Figure 5 depicts the parallelized encoding algorithm using static domain distribution scheme.

It can be seen that there are two kinds of tasks:**master task** and **slave task**. The master task involves the initialization, creation of initial range blocks, creation and distribution of domain pool, broadcasting range blocks, decomposition of range blocks and outputting encoded image. A slave task receives range blocks to be encoded, receives a subset of domain pool corresponding to its task

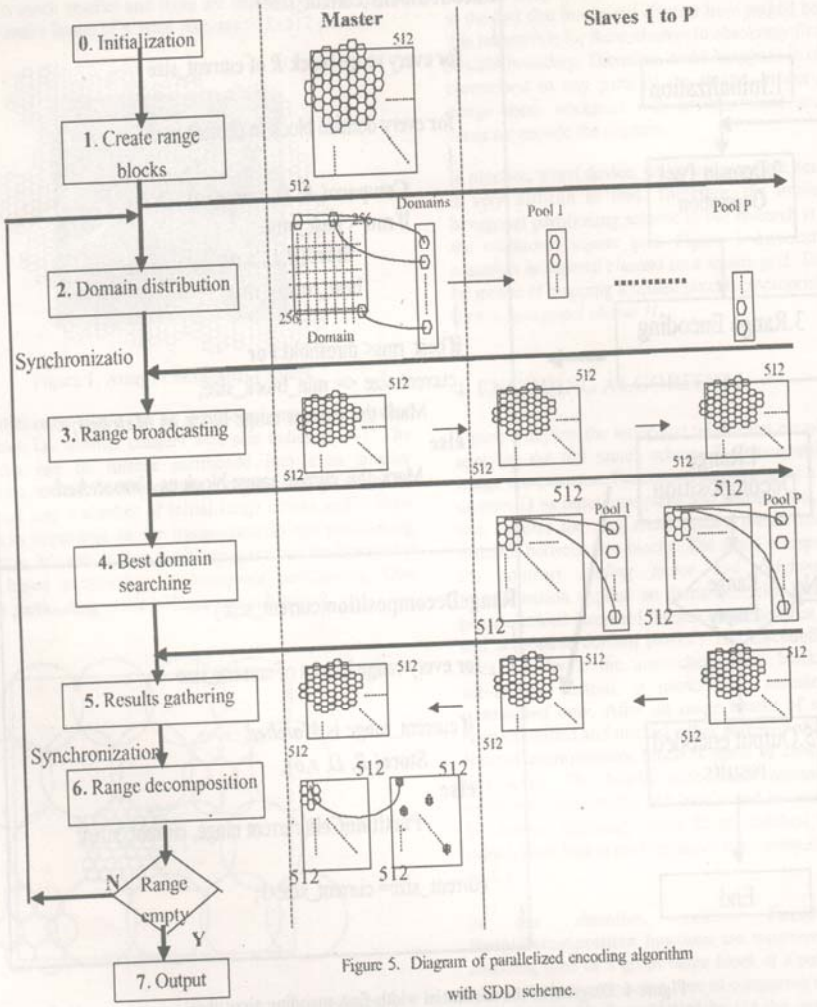


Figure 5. Diagram of parallelized encoding algorithm with SDD scheme.

Figure 6. Reconstruction image ('Lenna', 512x512) by HP(left) scheme (PSNR=29.07dB, comp. ratio=26.23) and, QT(right) scheme (PSNR=29.29dB, comp. ratio=26.04).

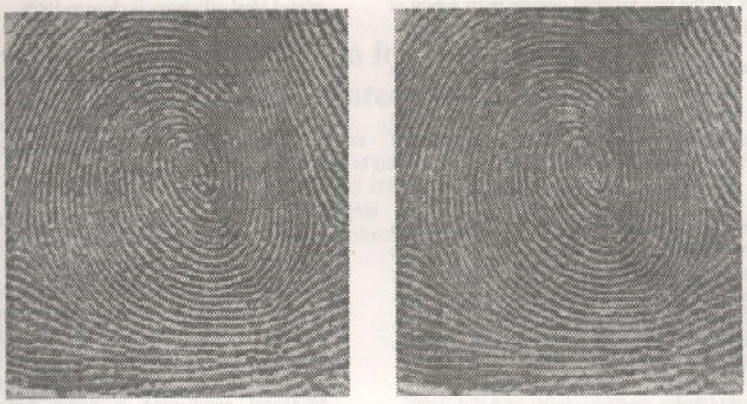


Figure 7. Reconstruction image ('Fingerprint s02', 256x256) by HP(left) scheme (PSNR=24.68 dB, comp. ratio=3.71) and. OT(right) scheme (PSNR=24.61 dB, comp. ratio=3.19).

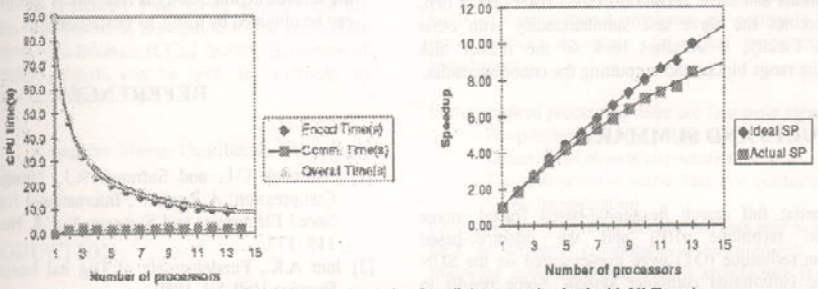


Figure 8. Encoding time and ideal & actual speedup for a light computing load with ULT mode.

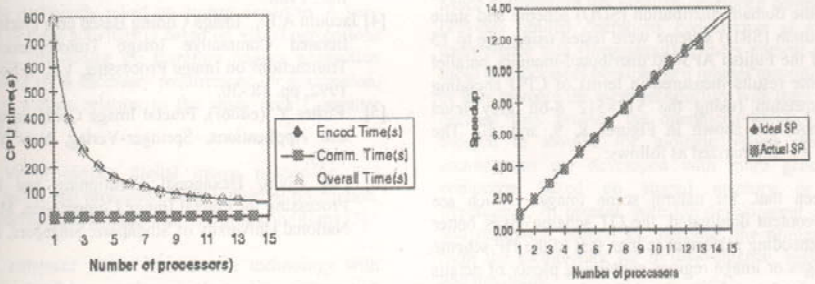


Figure 9. Encoding time and ideal & actual speedup for a medium computing load with ULT mode.

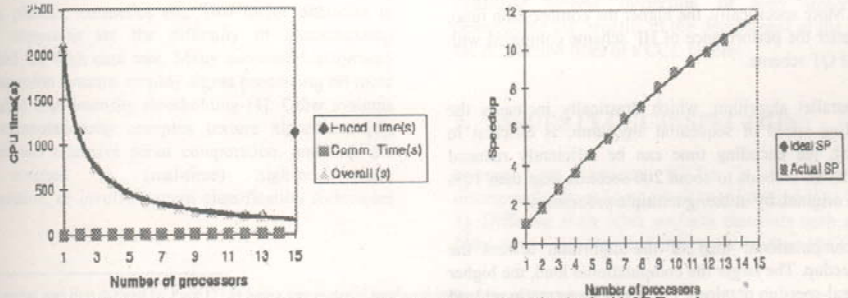


Figure 10. Encoding time and ideal & actual speedup for a heavy computing load with ULT mode.

number, searches for the best matched domain in its subset of domain pool and returns results to the master.

In this method, a master task creates the whole set of domain pool D and evenly divides the domain pool into non-overlapping subsets, then distributes these subsets among P different slaves. Different slaves receive their own subsets of domain pool Di and each subset has the same number of domain blocks.

On the other hand, the whole set of range cells are broadcast to all slaves and are encoded on every processor. Suppose there are Ni range blocks and Ki domain blocks in the whole pool at a particular iteration i and P slaves in execution. Then, each slave will have a private domain pool containing Ki /P entities. It means that the inner loop of function **EncodeBlocks** has been evenly partitioned on P processors.

One highlight is that during implementation of the algorithm, there is no need to assign a separate processor to serve as the master processor. We can assign one processor, say the *root* on a MPI (Message Passing Interface) machine, to function both as a master and slave. It can fulfill the master task at first, it then executes the slave task simultaneously with other processors. Finally, it switches back to the master task decomposing range blocks and outputting the encoded results.

6. RESULTS AND SUMMARY

The sequential full search hexagonal-based fractal image compression technique (HP) and the quadtree-based compression technique (QT) were implemented on the SUN Ultra Sparc Unix-based computer system. Some results in terms of Peak Signal-to-Noise Ratio (PSNR) and compression ratio are shown in Figures 6 and 7. The parallelized algorithms with both static domain distribution (SDD) scheme and static range distribution (SRD) scheme were tested using one to 15 processors of the Fujitsu AP3000 distributed-memory parallel machine. Some results measured in terms of CPU encoding times and speedup (using the 512×512 8-bit grey level "Lenna" image) are shown in Figures 8, 9, and, 10. The observations are summarized as follows:

(1) It is seen that, for natural scene images, which are smooth-content dominated, the QT scheme gives better overall encoding performance than that of the HP scheme. For images or image regions containing plenty of details and/or slanting edges, the HP scheme gives better reconstruction qualities both in terms of PSNR measure and the subjective visual evaluation, than the QT method does. More specifically, the higher the compression ratio, the better the performance of HP scheme compared with that of QT scheme.

(2) The parallel algorithm, which drastically increases the encoding speed of sequential algorithm, is efficient in general; the encoding time can be efficiently reduced from 2,080 seconds to about 200 seconds, less than 10% of the original, by utilizing multiple processors.

(3) The computational load of the algorithm affects the ideal-speedup. The larger the computational load, the higher is the ideal-speedup obtained. For heavy computational load applications, the ideal-speedup is almost linear and the related ideal-efficiency is close to 100%.

(4) The communication overhead makes the actual-speedup smaller than the ideal-speedup. However, for application with high computational load, the performance degradation caused by communication overhead is minor, this is especially true when the ULT (User Level Transfer) mode is employed.

(5) For applications targeting a reasonable compression ratio and reconstruction quality, say 26.23 and 29.07 dB, the computational load is so high that the ideal-speedup is almost linear and the effect of communication overhead is minor. Therefore, it can be concluded that the hexagonal partitioning based fractal encoding algorithm is suitable for parallelization. In general, experimental results show that the reconstruction quality for detail-dominant images using hexagonal partitioning is better than those of other partitioning schemes such as quadtree partitioning. Also, by running the parallelized encoding algorithm on multiple processors, the encoding time is drastically reduced while the reconstruction quality is retained. A speedup of about 10 can be obtained by using 13 processors.

REFERENCES

[1] Podilchuk C.I., and Safranek R.J., "Image and Video Compression: A Review", International Journal of High Speed Electronics and Systems, Vol. 8, No. 1, 1997, pp. 119~177.

[2] Jain A.K., Fundamentals of Dig ital Image Processing, Prentice-Hall, NJ, 1989.

[3] Barnsley M.F., Fractals Everywhere, Academic Press, Inc., 1988.

[4] Jacquin A.E., "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations", IEEE Transactions on Image Processing, Vol. 1, No. 1, January 1992, pp. 18~30.

[5] Fisher Y. (editor), Fractal Image Compression – Theory and Applications, Springer-Verlag, New York, USA, 1995.

[6] Fan Lixin, Hexagonal Partitioning and Its Parallel Processing for Fractal Image Compression, M.Sc. Thesis, National University of Singapore, Singapore, 1998.

An AOI System for Web-material Based on Distributed Network of PCs*

Song Peihua Gao Dunyue
College of Information
East China University of Science and technology
Shanghai 200237
E-mail: phsong@263.net

ABSTRACT

In this paper a distributed vision network is proposed to tackle industrial web materials inspection. The system consists of independent networked inspection PCs able to address efficiently parallel inspection tasks at a high production speed. Parallel algorithms for an automatic recognition and classification of defects of webs from an industrial line are presented. Finally, we also present the benefits of the deployment of the system in the production lines of a copper clad laminate (CCL) facility. By means of this classification, objects can be sent, for example, to different sectors of the line.

Keyword: Computer Vision, Distributed, Networked, Inspection, Online

1. INTRODUCTION

There is an increasing interest on handling and decision-making processes automation, which have so far been almost exclusively relied on human expertise. A very wide range of these activities is based on visual perceptions of the world surrounding us; therefore images acquisition and processing are a necessary requirement for automation, and there is a close relation to the study area Computer Vision [1] [2].

Computer vision applies digital image processing and analysis to tackle real problems in the industrial production, mainly of standardized products, in real-time conditions [3].

Nowadays, computer vision is a mature technology with numerous successful examples in electronics, automotive and pharmaceutical industry, while it penetrates slowly but steadily sectors such as full-surface detecting, textiles, ceramics, plastics, cosmetics etc. Two major obstacles to machine inspection are the difficulty of characterizing defects and the high data rate. Many successful automated visual inspection systems employ signal processing no more sophisticated than intensity thresholding [4]. Other systems involve computationally complex texture algorithms [5], which demand extensive serial computation, and thus are poorly matched to (real-time) high-performance implementation, or involve pattern classification techniques [6].

The purpose and logical structure of a computerized vision system is essentially the same as a human one. From an image caught by a sensor, all the necessary analyses and processes are carried out in order to recognize the image and the objects forming it. There are several considerations to be made when designing a vision system: What kind of information is to be extracted from the image? Which is the structure of this information in the image? Which "a priori" knowledge is needed to extract this information? Which kinds of computational processes are required? Which are the required structures for data representation and knowledge?

In the required processing there are four main aspects:
- Pre-processing of the image data
- Detection of objects characteristics
- Transformation of iconic data into symbolic data
- Scene interpretation

Each of these tasks requires different data representations, as well as different computational requirements. Thus the following question arises: Which architecture is needed to carry out the operations?

Much architecture has pipeline-developed computers, which provides limited operations concurrence at a good data transference rate (throughput). Others present a connected mesh of processors because their images mapping is efficient, or they increase the mesh architecture with trees or pyramids because they provide the operations hierarchy, which is thought to involve the biologic vision system. Some architectures are developed with more general parallel computers based on shared memory or connected hypercubes. At present, with the development of electronic and computer, general PC becomes the primary choice to form the architecture for its convenience, inexpensiveness and flexibility.

In this paper we present a distributed vision network for industrial defects inspection of web materials and demonstrate the benefits of the deployment of the system in the production lines of a CCL facility.

2. ISSUES IN CCL INSPECTION

Copper clad laminate (CCL) is homogeneous and discrepancies from homogeneity are interpreted as flaws (Fig. 1). Differing from other uniform materials such as metals, films, paper and various plastics, the surface of CCL is very

* A web-material was first defined by Purll [7] as being any material produced in the form of strips.
Textile, paper, glass, wood, metal, food, and industrial parts on a conveyor belt can all be cited under this heading.

bright like a mirror, which forms a specular reflection usually, but a diffuse reflection is our expectation. How can we avoid the specular reflection is discussed in another paper.

At present, there are commercially available systems that can perform defect detection in uniform web materials at a reasonable cost. However, the problem of defect classification is still an open research issue. The major obstacles in solving the classification problem are the following:

Extremely high data throughput. A typical web material is 1-3 m wide and moves with speeds ranging from 20 to 200 m min^{-1}. Consequently, data throughput for 100% inspection (when classifying defects of mm size) is tremendous and is difficult to be handled by the present general purpose hardware.

Inter-class similarity and intra-class diversity. A single class of defects may vary widely in appearance and have members that closely resemble defects in other classes. Therefore, the structure of a given class in a feature space may be of a very complex nature.

Large number of classes. A typical defect classification problem involves a large number of defect classes; it is not unusual to deal with a few dozen to a few hundred classes.

Dynamic defect populations. Small changes in the production process can result in entirely new classes of defects and a useful classification system should be dynamic with the ability for continuous on-line learning.

The first three items make initial system design very difficult, while the fourth item is a major obstacle in extending the useful lifespan of a developed system.

In order to illuminate data rates expected in CCL/web material inspection we consider a specific case. First, we analyze the problem of appropriate rate of data acquisition. For example, consider a 1.5 m wide web material which is moving at speed of 100 m min^{-1} and defects of width 0.2 mm (both in horizontal and vertical direction), and a defect must be represented by a minimum of 1 pixels in both horizontal and vertical directions (i.e. spatial resolution of 0.2 mm $pixel^{-1}$). It is necessary to place nine CCD cameras operating at 768 x 576 pixels to cover the cross web direction. In order to keep up with the moving web, which travels at rate of 1.67 ms^{-1}, it is necessary to acquire 20 frames s^{-1}. The defect detection subsystem receives in total 9 x 768 x 576 x 20=79 x 10^6 pixels s^{-1}.

Next we consider the problem of data processing and defect classification. The Ultimate objective of an inspection system is to process data in real time; however, the concept of real time may be understood differently, depending on the specific task. Generally, it implies that the processing can keep up with data acquisition and material manufacture. The simplest task in web material inspection is recording the position of each defect within the web map, so that this position may be taken into account at a later time, e.g. when cutting material.

3. GENERAL ARCHITECTURE

In order to keep up the data processing, the proposed system is a set of independent networked inspection PCs assigned to production lines (Fig. 2). A Server is connected via coaxial cable to the managing PC worked as Client forming an Ethernet backbone. Each Client has a unique IP address and can be connected to one camera. Each camera is installed in the production line with the appropriate lens and lighting equipment. The PC is a Pentium 733MHz Windows NT workstation or higher.

4. PROBLEM DESCRIPTIONS AND SEQUENTIAL SOLUTION

The application classifies the objects and makes decisions, which may be translated as signals, which send the objects to different sectors of the production line according to their characteristics, or discard them. For the classification process, different characteristics are considered, such as coordinates, area, width, height, even value of background, even value of foreground, max value of foreground, min value of foreground, chain code histogram (CCH) and the presence of defects on the surface.

As already mentioned, the goal is the automation of objects classification by using distributed algorithms in heterogeneous computers networks. For this purpose, a "master/slave" model was used. The master process is responsible of dividing processing activities among a certain number of slave or client processes, and of coordinating their calculation. Clients are in charge of performing the processing activities themselves.

There is parallelization at a data level: the image is partitioned in sub-images, which are captured by the clients (all of them identical) in charge of the processing activities. Some algorithms are easily parallelized, whereas for others special care must be taken as regards master participation in the processing of activities to coordinate clients. Each of the developed algorithms will now be analyzed.

4.1 Segmentation

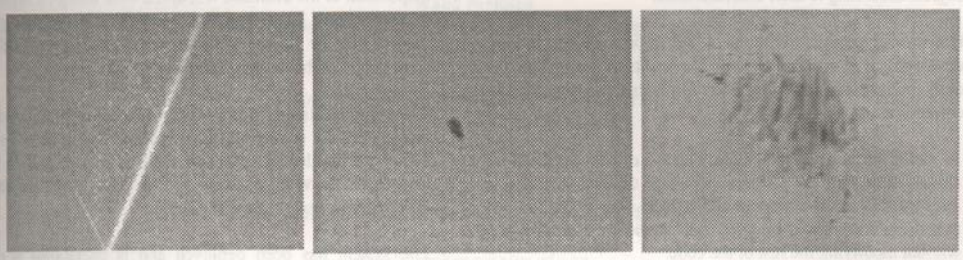
Threshold (a labeling operation on a gray-leveled or colored image) is one of the most widely used methods to segment a shape or a particular feature of interest from an image. The threshold binary operator produces a black and white image. But as a automatic operation we can not determine the threshold value properly, which may results in the following stages failed. So we segment the object by using the specific edge detection operator—Log-Prewitt that uses the same convolution kernels as Prewitt Operator, but the input is not the pixel values of image. The Log_Prewitt Operator uses the logarithms of pixel values of image as the input, which makes the result not sensitive for the variance of lightness. At the same time the Log_Prewitt Operator has inherited the smoothing performance of Prewitt Operator.

4.2 Connected Components Labeling

The implementation of a labeling algorithm would allow classifying several objects from one image, as well as to process control marks in order to, for example, determining objects size in a standard measurement unit. The labeling operator also generates a table of regions showing the amount of pixels belonging to each of the regions. By analyzing this table we can obtain information to determine

which is the biggest object in the image (the interest object), which are the very small objects representing noise or imperfections, and which are the objects of the pre-established size which in this case are control marks.

height, even-value of background, even-value of foreground, max-value of foreground, min-value of foreground and chain code histogram (CCH). It only goes through the images once as the better and fast algorithms to label connected



(a) Scratch (b) Oxidation (a) Cat-foot

Fig. 1 The sub-image as flaws captured by CCD online

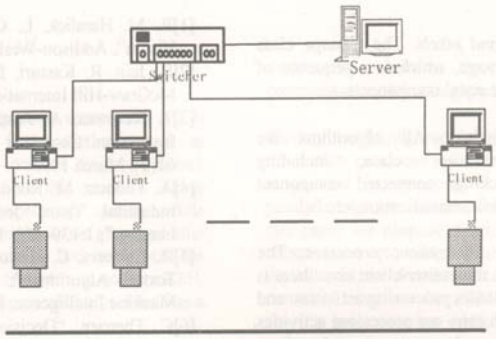


Fig. 2 The outline of the system

The algorithm consists on the labeling of the connected components of a binary image, followed by measurements over the regions found. Connected components labeling algorithms group together all pixels belonging to the same region and assign an only label to them (it is a grouping operation). The region is a more complex unit than the pixel, and presents a larger set of properties (shape, position, gray level statistics, etc.). One way of distinguishing defective objects or objects with different characteristics is to distinguish the different regions according to their properties.

The speed and storage cost of the algorithm executing this labeling operation is extremely important, and there are different variants to carry out this stage. Usually one row of the image is processed at a time, and new labels are assigned to the first level of each component, with attempts to extend the label of the pixel to its neighbors to the right and downward. But this will use large auxiliary storage to produce the labeled image

In our work, the boundary chain code was carry out to track the boundary of defects being segmented from the background to get features such as coordinates, size, width,

component.

4.3 Characterization Statistic

As pixels of the boundary are found, their coordinates and values can be got easily, thus the features such as coordinates, size, width, height, even-value of background, even-value of foreground, max-value of foreground, min-value of foreground and CCH is determined simply by counting the specific pixels.

4.4 Classification

One of the main objectives of the application is to determine the existence of defects on the object and classify them. To do this, an algorithm was developed to analyze the features identified by the labeling process and characterization statistic process.

5. DEVELOPMENT ENVIROMENT

The application was developed in C++ at Windows NT workstation, since it provides object-oriented programming possibilities, which facilitates the development of a clear, modular and extensible code.

As already mentioned, the application developed on a client-server architecture is oriented to become a process applied in real time on the images obtained with cameras, and results are sent to devices deciding the direction of the objects according to their characteristics.

However, a graphic environment had to be used during the development in order to observe the results of the images at each processing step in order to determine algorithms correction.

In order to abstract the main objects of the application, classes were defined. Classes were also defined in order to encapsulate calls to instances and process communication, so that this library could be later on changed by a similar one without modifying the rest of the code.

The encapsulating of algorithms as classes allows the addition of new algorithms in a very simple way and following a clear scheme, in addition of which it allows to generate lists of algorithms to be sequentially applied to an image.

Classes to abstract images and pixels. The *CImage* class encapsulates an gray scale image, which is a sequence of pixels, each of which has three equal components.

Classes to abstract algorithms. All algorithms are encapsulated in *CImageProcess* class, including segmentation, boundary tracking, connected component labeling, characterization statistic, classification, etc.

Classes to abstract master and client processes. The parallelism model chosen was the master-client one: there is a master process which coordinates processing activities, and several client processes, which carry out processing activities. This caused the definition of two classes to encapsulate these processes *(CMaster* and *CClient*). The general idea of the application is that an instance of an *CMaster* process is created, all necessary parameters are set (image to process, number of processes among which the processing activities will be distributed, etc.), and the processing task begins by calling a particular method of the object (process( )). This method instantiates the corresponding client processes according to the parameters, partitions the image among these processes and then coordinates their partial results in order to obtain the final result. On the other hand, client processes, when run, create an instance of the *CClient* class and begin the process by calling the process( ) method. This method receives the results, processes them, and synchronically exchanges them with the instantiating master process.

6. CONCLUSIONS

It would be convenient to evaluate separately the tools used and the results obtained. As regards the tools, Windows NT turned out to be very stable, In addition to this, it provides facilities for networks with a TCP/IP configuration (this characteristic is very important if we consider the orientation of the application). In short, environment selection proved to be suitable for an application with these characteristics (interconnection, concurrency, and handling of very large data volumes), in addition of which it allows to exploit portability.

The language used (C++) allows to define inheritance between classes and data encapsulation, which in turn allows to make a clear and easily extensible implementation; and due to its low level characteristic, it also allows an optimal disposition of the available resources, even though it requires excessive care when handling memory and the debugger provided by the version of the compiler used is non-friendly, error detection being thus complicated.

The results obtained during the performance measurement proved that the communications between Server and Clients is not the bottleneck although there is lots of data to be transferred between them, since a high speed Switcher (100M) to be used to form the LAN.

The system has been deployed and is operational in the production lines of a CCL-manufacture facility in an efficient and reliable way.

REFERENCES

[1]R. M. Haralick, L. G. Shapiro, "Computer and Robot Vision", Addison-Wesley Publishing Company, 1992.

[2]R. Jain, R. Kasturi, B. G. Schunck, "Machine Vision", McGraw-Hill International Editions,1995.

[3]T. Newman, A. Jain: A Survey of Automated Visual Inspection. Computer Vision and Image Understanding, 61(2), March 1995.

[4]A. Thomas, M. Rodd, J. Holt and C. Neill: "Real-Time Industrial Visual Inspection: A Review. Real-Time Imaging", 1:139-158, 1995.

[5]R. Conners, C. Harlow: "A Theoretical Comparison of Texture Algorithms", IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-2(3), May 1980.

[6]C. Therrien: "Decision, Estimation and Classification", Chichester, UK: J.Wiley and Sons,1989.

[7]D. J. Purll, "Solid State Image Sensors in Automated Visual Inspection", B. G. Batchelor, D. A. Hill and D. C. Hodgson, Eds., Elsevier Science Publishing Company, Inc., New York, NY, pp. 255-293, 1985.

Distributed Algorithm for Fractal Image Compression

Wang Meiqing
Department of Computer Science, Xi'an Jiaotong University
Xi'an, Shanxi, 710049, P.R.China
Or
Department of Mathematics, Fuzhou University
Fuzhou, Fujian, 350002, P.R.China
Email: mq_wang@163.com
And
Zheng Shouqi
Department of Computer Science, Xi'an Jiaotong University
Xi'an, Shanxi, 710049, P.R.China
Email: sqzheng@xjtu.edu.cn
And
Zheng Wenbo
Department of Computer Science and Technology, Fuzhou University
Fuzhou, Fujian, 350002, P.R.China

ABSTRACT

Fractal image compression has large time complexity and is restricted in a single machine. Some methods such as classification search method and nearest neighbor search method have been proposed to solve this problem. These methods depress the image compression quality while reducing the time complexity. In this paper we give a parallel algorithm for fractal image compression and implement the algorithm in distributed computing system base on Java RMI. The experiment results show that the parallel algorithm has a high speedup.

Keywords: Fractal, Image Compression, and Distributed Algorithm.

1. INTRODUCTION

Fractal image compression [1] builds on local self-similarity within images so it has a high compression rate and can be stored and transmitted easily. On the other hand, fractal image decode algorithm only need to compute the fixed point of an image operator, so it is very simple and suitable for the instances of once encoded and many times decoded. But, there are two primary problems for traditional fractal image compression. The first is that for some images, fractal compression can not achieve high quality; the second problem is that, while partitioning the image, the number of domain blocks used to match small range blocks is very large and the rate of fractal compression algorithms is very slow.

To solve large complexity problem some kinds of methods restrict the set of domain blocks so every search is in some region of the pool of domain blocks, for examples [2,3], classification search method, boundary-based distributing method, region-based matching method and nearest neighbor search method, etc. But these methods will increase the storage and reduced the compression rate. On the other hand, they will depress the quality of images because the matches are only in the local areas and maybe ignore the better matching regions. Recently Matthias Ruhl and Hannes Hartenstein consider in their study that it is a NP-hard problem to optimize fractal codes. So dealing fractal compressions with a single computer is limited.

Through analyzing traditional fractal algorithms we find that the search processes of the matching domain for all small ranges are independent, so these processes include data parallels and are suitable for distributed computing [4]. In this paper we propose distributed methods for basic fractal compression and adaptive quadtree method and implement the latter in the distributed computing system JDCS that we construct based on Java RMI [5,6]. The experiment results show a high speedup while not reducing the compression rate.

2. THE DISTRIBUTED COMPUTING SYSTEM JDCS BASED ON JAVA RMI

The distributed computing system JDCS based on Java RMI includes a monitor computer, a series of computing nodes maintained dynamically, and some clients. The monitor computer and all computing nodes compose a pool of servers. The role of the monitor computer is managing and maintaining the addresses and the ports of computing nodes, accepting the registry requests from the computers of the network, transmitting the computing interface program and server manage program to computing nodes. Any computers on networks can submit the registry request to the monitor, and become a computing node. The computing nodes implement the parallel parts of computing tasks. The clients are the host computers submitting the computing tasks. A client gets the addresses and the ports of computing nodes which can be used presently and transmit computing tasks to the computing nodes through the unique interface program, then the computing tasks can be run on the nodes. The architecture of JDCS is as illustrated in figure 1. From the point of Java RMI, all computing nodes are acted as servers of computing tasks relative to the clients. There can be many clients simultaneously in the JDCS.

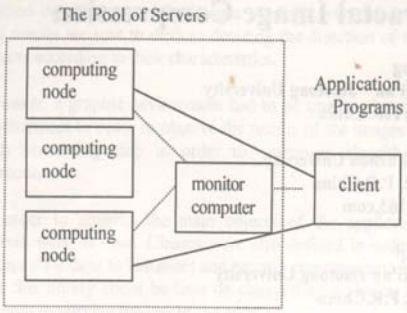


Fig 1. The Architecture of JDCS

3 THE DISTRIBUTED METHOD OF BASIC FRACTAL COMPRESSION

The fractal image compression of fixed size range blocks is the basis of other fractal compression, so we first analyze its distributed algorithm. The computing complexes of fractal image compression methods mostly lie in the searching optimize domain block procedures from the pool of domain blocks for every range block. In the distributed algorithm we regard these procedures as remote procedures and transfer them to the computing nodes of JDCS to carry out.

Suppose that the number of computing nodes which attend computing is N. The basic distributed fractal compression method is that:

1) Segment image in the client computer of JDCS.

Segment the given image using a fixed block size, e.g., 4×4. The resulting blocks are ranges R_i.

2) Create Domain block pool

By stepping through the image with a size of *l* pixels horizontally and vertically create a list of domain blocks from the image, which are twice the range size. By average the four pixels each shrink the domain blocks to match the size of ranges. The shrinking domains are called codebook blocks D_i.

3) The search

Create N threads in the client computer. Within every thread a connection with a computing node is created. Distribute equally all ranges to the N computing nodes.

In the nodes for range block R an optimal approximation $R \approx sD+oI$ is computed in the following steps:

- For each codebook block D_i, compute an optimal approximation $R \approx s\ D_i + o\ I$ yielding the least root-mean-square error $E(D_i, R)$;
- Find the block D_k with minimal error among all codebook blocks

$$E(R, D_k) = \min E(R, D_k).$$

- Transfer the codes for the current range block consisting of two compression coefficients s, o and the index *k* identifying the optimal codebook block D_k to the client computer.

4) In the client computer write the compression codes to the output compression file according to the sequence of range blocks.

In next section we will give the detail construction of the distributed algorithm for the adaptive quadtree method. The construction of other fractal algorithms such as rectangle partition or triangle partition is similar.

4. THE DISTRIBUTED ALGORITHM OF THE ADAPTIVE QUADTREE METHOD

4.1 Sequential algorithm

First we describe the sequential adaptive quadtree method given by Fisher:

1) Define a tolerance tol for the root-mean-square error, a minimal range size rmin and a maximal range size rmax. Partition the image into ranges of rmax × rmax size.

2) Initialize a stack of ranges by pushing the maximal size ranges onto it.

3) While the stack is nonempty carry out the following steps:
   a) Pop a range block R from the stack and search the corresponding codebook yielding an optimal approximation $R \approx sD+oI$ and a least rms error E(D,R).
   b) If the root-mean-square error is less than the tolerance or if the range size is equal to the minimum range size, then save the code for the range, i.e., s, o, and address of D (d_x, d_y). If s=0, do not store the rest.
   c) Otherwise partition R into four quadrants and push them onto the stack.

The main work of the adaptive quadtree method lies in the adaptive quartered processes. So the distributed implementation of the whole algorithm is mainly distributed implementation of adaptive quartered processes.

The quartered processes of the method partition the image blocks recursively and search the optimal domain block for every range block. There are data parallels in them. But the method need to write the compression codes to the same output file and is I/O dependent. To remove the dependence we store the compression codes of every parallel block to different output files and after the end of compression compose these files to a total output file according the order of blocks.

4.2 The distributed method of the adaptive quartered processes

Suppose the number of the computing nodes presently is N. In order to run the method in many computing nodes, it is needed to spread the recursive process manually and decompose the recursive process to a circulation. In the circulation a new recursive process begins. The circulation

distributes the new recursive process to the computing nodes.

If the image blocks that initially call the quartered process are far bigger than the most range size rmax, then we partition the image blocks into four quadrants manually. The number of computing nodes N decides the partition times. The spare partition process will be continued in the computing nodes. Because the remote method invocations are distributed to the computing nodes through different threads in the clients, the running threads become more as N becomes more. This will depress the performances of the clients. When the original images are far bigger than the most range size ramx, partition the image once gets four range blocks, partition twice gets 16 range blocks, and partition three times will get 64 range blocks. If the 64 threads are yielded and every thread will build a connection with a different computing node simultaneously in a same client, the client system would depress the information seriously. In practice we only get 16 nodes at most once. Thus the distributed method of the quartered process first partition the initial image into four quadrants twice and gets 16 range blocks. Then we create a group of threads in the client and yield a thread for every range block. Every thread builds a connection with a different computing node and distributes the match search process of the range block to the node and run.

The distributed method includes the remote process ParQuadtree and the client program CliQuadtree. The remote process ParQuadtree is used to quadtree the range block on the computing node. And the client program CliQuadtree presides over building connections with all nodes and distributing the process ParQuadtree to the nodes to run.

As similar to the sequential algorithm, the remote process ParQuadtree is also a recursive process. The difference is that the number of partition times is less 2 than the sequential one.

The client program CliQuadtree distributes ParQuadtree and the range blocks equally to the computing nodes. Suppose the number of the computing nodes is N, and define the minimum of N and 16 as M. CliQuadtree is as follows:

(1) Partition the initial image into four quadrants twice and get 16 range blocks;
(2) Create a thread and start the thread for every of the M nodes;
(3) Every thread carries out the following steps:

- Create a Security manager;
- Search the computing node and the server program running on the node relative to the thread;
- The computing node implement the parallel quadtree process ParQuadtree whose parameters are *i*, *i*+M(when N<16), ..., through calling the remote interface provided by the server program and write the compression codes to the file Pout[*i*], Pout[*i*+M], ..., .
- Return the Pout[*i*], Pout[*i*+M] ,...., to the client computer.

(4) Combine all files such as Pout[*i*] to the total output file according to the order of the range blocks.

4.3 The distributed method with tolerance

The distributed system JDCS is composed of the computers connected by the Intranet or Internet, the client programs may fail to connect with the computing nodes. That is, the search for the remote interface or calling the remote process may fail. To avoid these cases we modify the client programs using the exception catching mechanism (try/catch) provided by Java language so that the client programs can transmit the computing tasks to another node when the relative computing node occurs exceptions. The method is setting failure tags for the tasks on the failure nodes and for the failure nodes themselves. Define integer m1 as the number of complete tasks presently and integer m2 as the number of failure tasks presently. If the client computer catches a exception when it build a connection with the node *i*, we set the node *i* a failure tag; If the remote process *j* has a exception while implementing we set the remote process *j* a failure tag and return it to the client program. And the client redistributes it to another natural node. m1=16 denotes all tasks complete successfully.

4.4 The distributed implementation of adaptive quadtree method

Now we can describe the full distributed adaptive quadtree algorithm according to the distributed method of adaptive quartered processes:

1) Run the sequential part on the client: input the original image and all parameters needed by fractal compression; compute the pixels of codebook blocks by averaging four neighbor pixels; and classify the all domain blocks probably come forth.
2) Partition the image to maximal square blocks.
3) Partition every block to 16 equal range blocks.
4) Yield M threads (if N>16 then M=16 else M=N), distribute equally the quardtree search processes for 16 range blocks to the all computing nodes through remote procedure class ParQuadtree, and then receive the compression files Pout[*i*], and finally combine all files such as Pout[*i*] to the total output file according to the order of the range blocks.

The, flow of Distributed adaptive quadtree algorithm is as illustrated in figure 2:

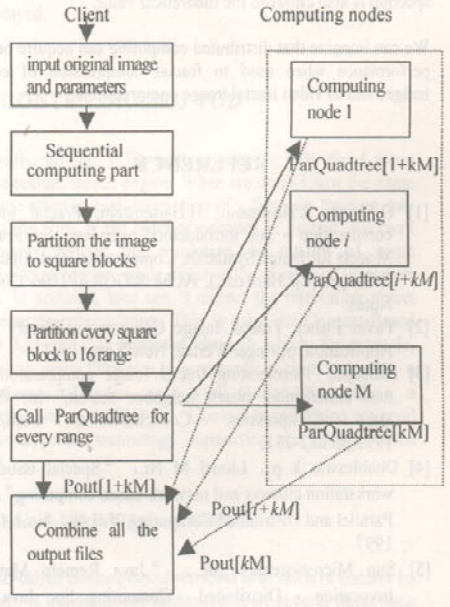


Fig.2 The flow of Distributed adaptive quadtree algorithm

5. THE EXPERIMENTAL RESULTS AND CONCLUSIONS

For the sake of simplexes, we suppose the run speed of client is the same as that of the computing nodes, and we don't consider the communication time delay between the client and the nodes. For the original 256×256 image, if the number of nodes is 4, the theoretical value of speedup is about 3.98. For the original 512×512 image, if the number of nodes is 8, the theoretical value of speedup is about 7.9.

We have implemented the distributed adaptive quadtree algorithm on the distributed computing system JDCS. On the experiments the JDCS system is composed of a monitor computer and four computing nodes. All computers are based on PIII CPU、64M memory and 15G HD. The operation system we used is Windows 98, and the version of Java is 1.2.2. The tolerance of rms error is 8.0, and the search is in all classes and subclasses.

When the original images is 256× 256 Lena image with 8 grey level, the run time of the sequential algorithm is about 12 seconds, and the run time of distributed algorithm in JDCS is about 7 seconds. The speedup is about 1.71.

When the original images is 512× 512 Lena image with 8 grey level, the run time of the sequential algorithm is about 52 seconds, and the run time of distributed algorithm in JDCS is about 17 seconds. The speedup is about 3.05.

It can be found that the speedup is closer to the theoretical value when the computing data is more. The time used to build connections between the client and the computing nodes is fixed, and the data stream transferred in the computing processes is also stable. So the more run time is, the less rate of network time delay to run time is, and the speedup is also closer to the theoretical value.

We can imagine that distributed computing can acquire better performance when used to fractal compression of color images and to video fractal image compressions.

REFERENCE

[1] D.Saupe, R.Hamzaoui, H.Hartenstein, Fractal image compression - An introductory overview, in: Fractal Models for Image Synthesis, Compression, and Analysis, D.Saupe, [2]J.Hart(eds.), ACM SIGGRAPH'96 Course Notes.

[2] Yuval Fisher, Fractal Image Compression-Theory and Application, Springer-Verlag, New York, 1994.

[3] D.Saupe, "Accelerating fractal image compression by muti-dimensional nearst neighbor search," In: IEEE Data Compression Conf,Snowbird Utah,Mar 1995,pp.222-231.

[4] Dhableswar k p.，Lionel M Ni.，"Special issue on workstation clusters and network-based computing"，J. Parallel and Distributed Computing, Vol.40，No.1:1-3，1997

[5] Sun Microsystems，Inc.，"Java Remote Method Invocation - Distributed Computing For Java"，http://java.sun.com/java/Sun Microsystems，Inc.，"Java Remote Method Invocation Specification"，http://java.sun.com/products/jdk1.2/docs/guide/rmi/，Sun Micro systems ,1999.

The Content-based VOD System

Jia Zhentang, Li Lingjuan, He Guiming
Integrated Media Research Center , Wuhan University
Wuhan , Hubei Province, China
Email: j_z_t@263.net

ABSTACT

In this paper, the concept of content-based VOD is proposed, which offers content-based search and transmission as well as object-oriented manipulation. The architecture, relevant techniques and problems are discussed. In the last section, part of our research work on the relevant techniques is presented.

Keywords: VOD, Content-Based Video Retrieval, Video Compression, Scalability, Video-Object, Video Segment

1. TRADITIONAL VIDEO-ON DEMAND (VOD)

VOD is an interactive "television" mode. User can select programs and then control the playing progress (play, fast-forward, reverse, pause, stop, etc.) [10]. Users are active but not passive as the traditional TV constrained. VOD can provides a wide range of service, such as film, news, weather report, music, education, archives record etc. VOD is mainly based on three technologies: video/audio compression, Video database and network communication. The basic structure is illustrated in Figure 1 below.

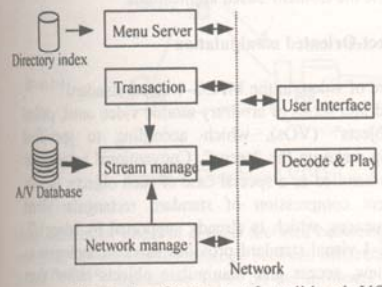


Figure 1. Structure of traditional VOD

Server Side:

Program Menu

The server provides clients a program menu, usually with a search function. This service is implemented in the form of web-page by a web server working with the VOD server. It means that the whole procedure begins from a web browser.

Transaction processing

Information of the selected program is sent to the transaction processing module, which will check the right of user, the amount of available resource on server, and the state of specified film, and then decide whether permits user's demand or temporarily puts user into the waiting queue (some modules are not showed in the Figure for the purpose of clarity). User's play-controlling commands are also interpreted here.

Stream management

Allocates memory and disk buffer, retrieves Video(and audio) data from video database, and streamingly sends out. It will adjust the sending bit-rate according to the situation of network. It can be seen as a Media-streaming-pump and its manager.

Network management

Check and report the situation of network. Some protocol designed for real-time media transmission will be used: RTP, RTCP, RTSP and RSVP. Some special Techniques is also used for specified application, such as "Sure Stream" for Real-system, "Intelligent Stream" for Microsoft Media Technology.

Client Side:

What users can do is just to select a program from the presented menu, and then the received A/V data is decoded and played.

2. CONTENT-BASED VOD

Currently, all the VOD systems provide only fixed menu with a textual search engine. When we don't know the name or some other textual description of a film, we have no way to get the film. If the band of network is limited, we can't selectively reduce the resolution or quality of some insignificant content while keeping the quality of what care more. In addition, user can 't extract the interesting object for separate manipulation. This is why the Content-Based VOD (CB-VOD) is proposed, through which, for example, Users can query films just by an sample, such as an image, a few frames of video ,or a VOP (Video-Object-Plane, a concept used in MPEG-4). Content-based video retrieval [2-6] is a prime technology supporting the content-based VOD.

2.1 Content-based Video retrieval

The so-called content-based retrieval is to retrieve content by content. It integrates the technologies of several fields such as image processing, pattern recognize, computer vision and DBMS. Content-based retrieval is to compare the given

image (or a section of video, as a sample) with the stored Video. It is a direct comparison between images or image features (of color、texture、shape、size, etc.), which needn't be represented in semantic level, so can provide direct、efficient and flexible retrieval. As illustrated in figure 2 .

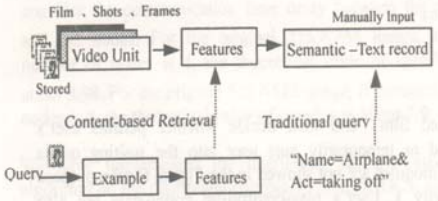


Figure 2. Content-based

In order to achieve content-based fast retrieval, Audio/Video data must be processed before retrieval, including video structure analysis and video unit automatic indexing [2,5,7]. Structure analysis is to detect the boundary of video shots and to divide video into a series of shots and then organize shots into scenes. Automatic indexing is to find the representative frame ,static and dynamic feature of shots, so as to form a feature space of shot and finally form clustering for semantic description.

It is obviously a hierarchical structure: film---scene ---shot---representative frame, with each layer having its specified properties. The film (file) has the global properties, such as type, title, abstract, producer, director, and actors, which are the main information for traditional query. The features of shots and frames can be divided into two classes: static, and dynamic.

Static features: mainly come from color 、shape and texture [1]; Motion analysis: motion can be divided into two different kinds: Camera movement and object movement, (or both of them), with each kind having different character respectively.

2.2 Content-based VOD

The structure of CB-VOD is illustrated in Figure 3. below.

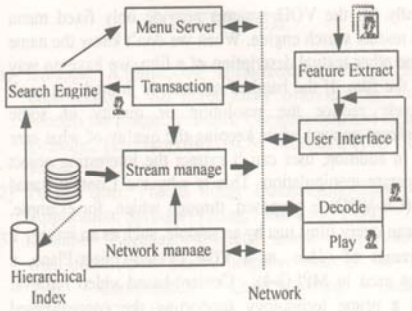


Figure 3. Structure of content-based

A Content-based VOD should support the following functions:

Content-based query User can query films by sample picture or simple sequence as well as text description. Usually the features of sample (not the sample itself) are transmitted to server for search, so there should be a Feature-Extract module in the client side. A sample picture (or sequence) may be input by user or pasted from current stream.

Content-based search engine In order to support the content-based query, a content-based search engine should be built up. And the content-based video retrieval is a core technique. The search engine accepts the features of sample, searches them in the feature space of stored video, and provides a list of result sorted by similarity. The menu server can dynamically refresh menu-pages according to the result produced by search-engine.

Content-based transmission The content-based scalability should be supported. The important content or what the user interested in should be transmitted in high quality while the less important are not transmitted or transmitted in low quality.

Content-based manipulation User can freeze the current screen and select any video object that are presented in the displaying-window, and then copy to clipboard, save to disk, paste to the query window, or do other operations.

The traditional frame-based features can't accurately depict the features of video-object, instead it only represent something of entire frame. Shape, for example, is an essential feature for video object. It is certain that shape can be employed to improve the accuracy and efficiency of retrieval. Shape is related to a certain object, so the object must be extracted from background before its shape can be coded or be compared, and this is beyond the current content-based retrieval.

Along with Mpeg-4, the concept of Video-Object is proposed and the object-oriented coding and manipulation is adopted as a main part of Mpeg-4. It will certainly extend and promote the Content-based applications.

2.3 Object-Oriented manipulation

The content of video, in the MPEG-4 visual standard [15,16,18], are divided into series of arbitrary atomic video units, called "video objects" (VOs), which according to specified relationship make up the "scene". Conventional rectangular imagery is handled as a special case of such objects. Beside the efficient compression of standard rectangular sized image sequences, which is already supported by Mpeg1/2, the MPEG-4 visual standard provides users technologies to encode, view, access and manipulate objects rather than pixels.

Each Video-Object can be manipulated individually. The shape (including size), motion and texture information of the

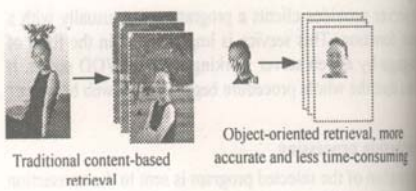


Figure 4. Object-oriented retrieval

VOPs (Video Object Plane, temporal slice of VO) belonging to the same VO is encoded and transmitted together in a certain VOL (Video Object Layer), while scene information that describes the relation of VOPs is also included in the bit-stream. The stream provides ready information for Object-Oriented retrieval, so the search sphere become small and it can be completed in a short time, accurately. Moreover, users in VOD system can freeze the picture and then extract any interesting VO (a human-face, for example) to use as sample for further objected-oriented query.

In addition, a large range of scalability is supported by Mpeg-4. We can identify and selectively transmit the video object of interest or of most importance. In the case of live broadcast, the interesting (or important) objects may be assigned more bits. Video objects can be transmitted in multiple layers, such as the base-layer and the enhancement-layer, through which both spatial scalability and temporal scalability are implemented. For spatial scalability, the enhancement-layer improves upon the spatial resolution of a VOP provided by the base-layer; In temporal scalability, the frame (or VOP) rate of the video is enhanced.

3.DISTRIBUTED ENVIRONMENT

The number of concurrent streams provided by one server can't overstep a threshold that is determined by: the power of computer, the available network band, and the bit-rate of each stream. A mass system is usually divided into several small ones, as illustrated in Figure 5. (We call this structure "Video Chain-Shop"). In each region, the storage is usually hierarchical: stream server, archive server, tertiary server. Each server may also comprise more than one computer.

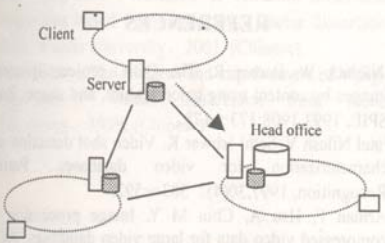


Figure 5. Distributed VOD

Here, we are not intent to discuss the load balancing among servers or the duplicating strategy between different storage levels [9,12,13,14], instead we focus on the influence imposed by content-based functionalities.

(1) It is not the sample itself but the features of sample that needs to be transmitted. It can be transmitted in up channel, consuming less network resource.

(2) But the content-based search is time consuming, the optimized search strategy should be applied:
- The search link should not go too far. Firstly resort to the chain store that has high probability of success and high connection bit-rate.
- Send back a menu firstly. Then, if demanded by user, a whole film is duplicated (from the specified chain store).
- The hierarchical index information related to the film should also be duplicated.

(3) The object-oriented transmission provides more scalabilities, based on which some new efficient algorithm for data exchanging may be developed.

(4) There should be a standardized interface: How to represent the features of video, and how to exchange information between client and server or between two servers. It can refer to the definitions in Mpeg-7. Mpeg-7 standard, which is known as "Multimedia Content Description Interface", aims at providing standardized core technologies allowing description of audiovisual data content in multimedia environments.

4. OUR RESEARCH WORK

By 1996, we had developed ourselves' Mpeg-2 and H.263 encoder/decoder [23]. Based on these algorithms, a series of applications were developed, including Videophone, Videoconference, Video Monitor, and Video-on-demand. At the same time, content-based Video retrieval was also under research, but before 1997, this work is mainly based on uncompressed video or Image. From 1997, we begin to put our hand to compressed data. From 1998, after the object-oriented concept is proposed in Mpeg-4 draft, We begin to conceive the content-based VOD system which combines the VOD technologies and the technologies of content-based video retrieval.

Based on the ready technologies in VOD [21] and video retrieval, we constructed a model system, which runs in IP LAN (via UDP, RTP, RTCP, RTSP protocols). For convenience, the web browser is not used, instead we provide user a video browser specially developed with VC++.

Content-based VOD is a comprehensive system that relates to various technologies, including video storage and management, video indexing and retrieval, streaming media transmission, and video/audio compression, especially object-oriented encoding. Most of these technologies are immature and in development. So, it is hard for us to build a complete and high efficient system. We are being engaged in the research of the relevant key technologies, and trying to make it a practical system

About content-based video retrieval

We had built up a test Video Retrieval system based on dominant color extraction and dominant color matching of the representative frames. This system runs on uncompressed video. Recent years, we pay more attention to the retrieval methods on compressed video.

The present compression standards, such as Mpeg-2 and H.263, provide limited clue for retrieval, including the DC coefficient of I-frame as average intensity, AC coefficient as a measure of texture, and the motion vector of P-frame or B-frame as a measure of movement. In order to create index conveniently, we modified the present H263 encoder and constructed a new platform, which can create index system while compressing video stream, as shown in Figure 6.

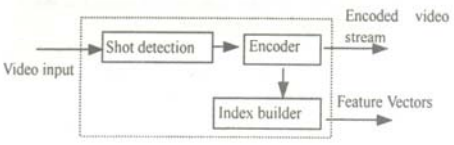


Figure 6. Modified encoder

Shot detection is a "enhanced motion detection", which can detect the gradual change of scene as well as abrupt change. Intra-frame coding mode is adopted for the first frame of each shot. In order to reduce computational complexity, we use a block-based method for shot detection. No more than three representative frames are selected for each shot, and two feature vectors are determined: dominate colors and a nine-dimension motion vector. The output video stream is same as the normal compressed video data.

We also tried to index video by the text information embedded in video pictures, and a novel stack-based text event matching algorithm was proposed to improve the accuracy of text detection [23].

About video compression

Video compression is a key technology, with different compression algorithm determining different retrieval and transmission method. We use H.263 in our early work, which is for very low bit-rate application, with simple algorithm and high compression rate. But such frame-based method can provide only global information.

Now we are aiming at the object-oriented compression method.

For object-oriented compression, the key technique is the extraction and tracking of video object. Both spatial and temporal information should be utilized simultaneously. For spatial information, we can use parallel region growing directly on intensity according certain consistency criterion [20], or use watershed algorithm on gradient image [19]. The distribution of color and texture can also be considered. For the temporal (Inter-frame) information, there are three modes: optical flow, motion Vector (usually block-based), and displaced frame difference (DFD). They are essentially accordant. The optical flow need complex computation, Motion vector is more coarse but fast, and DFD is the simplest.

The algorithm under our testing now employs the DFD, motion vector and spatial gradient. The DFD between two continuous frames are calculated firstly (using the fourth-order-moment and Opening-Closing morphological filter), then the vectors are used to eliminate the uncovered background. (We also tried "the intersection of two continuous DFD " to get the VOP mask, but it is not satisfying). Finally a relaxation is made according to the gradient information to get more accuracy border. During the process of segmentation, a vop buffer is maintained, through which a proper region is predicted for an previous vop when do vop matching. Both the spatial texture and motion vector determine whether each part of the buffer should be updated or not.

Our current test are constrained in a simplified situation: Only translation motion (tx, ty) is assumed for background Sprite coding ; the motion of each region is described by a simplified linear model where four motion parameters need to be estimated, which are related to zooming, rotating, and translation , respectively; And only the moving (or once moved) regions are considered as objects while all the still parts are taken as background.

Next step, we will do further study on vop shapes for their parametric description , tracking, and encoding. A more complex motion model will also be adopted for background motion estimation.

5. CONCLUSION

In this paper, a content-based VOD framework is proposed, which emphasize three techniques: traditional VOD (media storage and manager, streaming transmission), content-based video retrieval, and video compression. In order to efficiently implement content-based process, the object-oriented compression should be adopted.

It is a hard work to implement a satisfying content-based VOD, but no doubt it is a trend. The difficulties are in that:

- The speed and accuracy of content-based retrieval are not satisfying.
- The algorithms of object-oriented segmentation, compression and manipulation are still immature.
- The technology of compression-domain-based analysis needs to be further studied.

We think, with the new standards (Mpeg-4, Mpeg-7 and Mpeg-21) being put into practice, content-based VOD will be greatly developed.

REFERENCES

[1]. Niblack W, Barber R. The QBIC project: querying images by content using color, texture, and shape. Proc. SPIE, 1993,1908:173～178.

[2] Patel Nilesh V, Sethi Ishwar K. Video shot detection and characterization for video databases. Pattern Recognition, 1997,30(4): 583～592.

[3] Arman F, Hsu A, Chiu M Y. Image processing on compressed video data for large video databases. ACM Multimedia, 1993,267～272.

[4] Suresh K Choubey, Vijay V Raghavan. Generic and fully automatic content-based image retrieval using color. Pattern Recognition Letters, 1997,18:1233～1240.

[5] Zhang H J, Wu Jianhua, Zhong Di et al. An integrated system for content-based video retrieval and browsing. Pattern Recognition, 1997, 30(4):643～657.

[6] Edoardo Ardizzone, Marcola Cascia. Automatic video database and retrieval. Multimedia Tools and Applications, 1997,4:29～56.

[7] BAI Xuesheng, XU Guangyou, Content based image retrieval and related techniques, ROBOT, Vol.19, No.3 May, 1997 (Chinese).

[8] Banu Ozden, Rajeev Rastogi, Avi Silberschatz. On the design of a low-cost video-on-demand storage system. Multimedia System, 1 996, 4(1) :40～ 54

[9] Brubeck D W, Rowe L A. Hierarchical storage management in a distributed VOD system. IEEE Multimedia, 1 996, 3 (3) : 3 7～ 47

[10] Wang Jian-song, The system analysis and key techniques of VOD, "Television Techniques", Vol.5, 1997 (Chinese)
[11] Liu Heng-Zhu, et al, The improvement on proportional servicing policy in video server, Chinese J. Computers Vol. 22, No. 2, Feb. 1999
[12] Li Yong, et al. Design and Management of Large Scale Hierarchical VOD Storage System, JOURNAL OF SOFTWARE, Vol. 10, No. 4 Apr. 1999 (Chinese)
[13] Meng Gui-e, The design of hierarchical extensible distributed VOD, electronic technique, Vol. 8, 2000 (Chinese)
[14] PENG Yu-Xing, et al. Distributed VOD System and Its Data Storage, CHINESE J. COMPUTERS, V23 .No 6, June 2000
[15] Overview of the MPEG-4 Standard ISO/IEC JTC1/SC29/WG11 N3444
[16] MPEG-4 Natural Video Coding - An overview Touradj Ebrahimi, Caspar Horne
[17] MPEG-4 Requirements ISO/IEC JTC1/SC29/WG11 N3154 December 1999 Maui
[18] Leonardo Chiariglione MPEG and Multimedia Communications IEEE trans CSVT, VOL. 7, NO. 1, FEBRUARY 1997 5
[19] Demin Wang, Unsupervised Video Segmentation Based on Watersheds and Temporal Tracking , IEEE Trans on CSVT VOL. 8, NO. 5, SEPTEMBER 1998 539
[20] Lu Guan-ming, Region-growing watershed algorithm and it's application in video segment, Journal of Nan-Jing university of Post and telecommunications (Natural Science), Vol. 20, No. 3 (Chinese)
[21] Zhang Qi, The research of VOD and video server, Master dissertation from Wuhan University 1998 , (Chinese)
[22] Hu Hong-bin, Research of semantic information extraction based video indexing, Doctor dissertation from Wuhan University, 2001, (Chinese)
[23] Chen Chen, The research and improvement of Mpeg-2 algorithm , Master dissertation from Wuhan University, 1999, (Chinese)

System Architecture for Digital TV Assembly Edit And Broadcast Control High Speed Wide band Network

Zhang Bixiong
Department of Computer Science, Wuhan
University of Technology, Wuhan 430063, P. R. China
Email: zhbxong@public.wh.hb.cn

ABSTRACT

This paper presents a plan of digital TV assembly edit and broadcast controlled by high-speed wide band network. Because it is uses the parallel distribution scheme, multi-channel digital TV programs are real-time in parallel assembly editing and in parallel broadcasting with high-speed code stream. Besides, many services for Video-On-Demand (VOD) and for public information (Stock, E-Business, Transportation, Education and Internet) are provided.

Keywords: Digital TV, Assembly Edit and Broadcast Control, High speed, Wide band network, VOD

1. INTRODUTION

How to design, edit and broadcast, and control Digital TV programmes, and how Digital TV programmes serve for VOD and public information (Stock, E-Business, Transportation, Education and Internet) are the questions which belong to a high speed developing field in the world (such as HDTV of USA, DVB/DVC/DVS of Europe etc.).On Oct.1st 1999,the first broadcast of HDTV was presented in Beijing of China. Nowadays many TV Stations are arranging digital TV in China.

Digital TV provides high quality video and audio (multi-channel surround stereo sound), besides many services for VOD and public information. However, it also produces many difficulties.

A great quantity of data and a high-speed Real-time video code streaming of digital TV, challenges computer networks to store and transmit data.

Because common computer networks don't satisfy the above needs, many new technologies were developed. How to use the new technologies to design real digital video network systems is based on the status of a TV Station, are the subjects of this paper.

The operations of a TV Station are generally divided into some parts as follows: programme photography (including news, arts, physical culture, theatre, music, science, education, economic and advertisement...etc.), programme editing (including programming and examine), programme assembly editing, broadcast control and transmit management. Every part and its corresponding departments are special and independent. Thus it's important for planning digital TV systems that video LAN of the linking departments must be designed before the main video network will be planned.

The plan presented in this paper is for a high-speed wide band Fiber Channel (FC) video local network that links department of Programme Assembly Edit with department of Broadcast Control.

2. ARCHITACTURE AND DETAILS

Fig.1 presents the main architecture, in which the left of the dotted line is the part of Assembly Editing and the right is the part of Broadcast Control.

This is a plan of directly connecting a nonlinear assembly edit network to a linear broadcast control network and it is also the primary development in China.

This plan is different with the common plan of independent assembly editing and broadcast by Disk Broadcast Server. In our plan the assembly edit station can send commands to modify the programme table and then send an inserting or switching programme with a corresponding table of modifying programme to the Video Server. This will occur after programme has been sent to Broadcast Control and before it is to be broadcasted by Video Server, so Broadcast Control can smoothly and automatically link the inserting or switching programme.

When the assembly edit station modifies programme table and downloads programme, some parallel algorithms must be used to rapidly cut or delay original programme according to its character for preventing cutting from the head or tail of programme section to be broadcasted.

2.1 Assembly Edit

Assembly Edit is an oriented store Fiber Channel (FC) network, which consists of an 8 ports FC switch, a FC hard disk array, 6 assembly edit stations and a FC server. The Assembly Edit stations finish programme assembly edit of 6 channels of CATV (every channel broadcasts 20-24 hours a day), in which programmes of 2 channels are transmitted to wireless transmitter. All assembly edit stations share FC hard disk array by FC switch. At the same time each assembly edit station is an up/download station of programme, therefore Dayang 3000 LE is chosen. On one hand, assembly edit is frame editing of linking programme, so the compressed format of editing programme to be used is M-JPEG format (it is better than MPEG-2 I frame and MPEG-2 IBP frame format, such as better quality of video, more expediently and smoothly linking frame of video). On the other hand, assembly edit is the last editing before broadcast, so a 4:1 ratio is chosen as the compression ratio. In order to satisfy the storage of 6 channels, we choose the FC hard disk array (180G×6), which is used as an assembly edit video database.

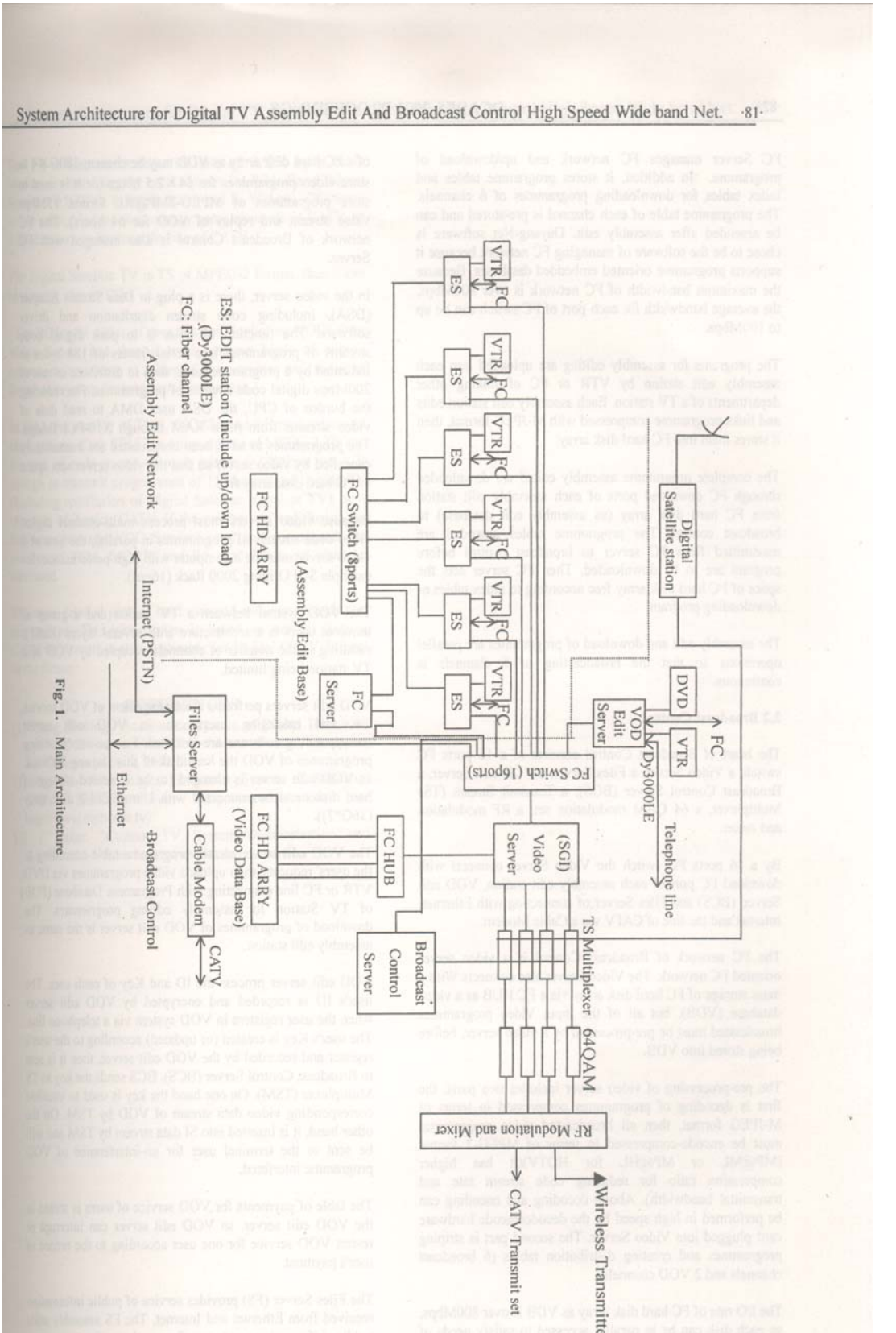


Fig. 1. Main Architecture

FC Server manages FC network and up/download of programme. In addition, it stores programme tables and index tables for downloading programmes of 6 channels. The programme table of each channel is pre-stored and can be amended after assembly edit. Dayang-Net software is chose to be the software of managing FC network because it supports programme oriented embedded databases. Because the maximum bandwidth of FC network is over 800Mbps, the average bandwidth for each port of FC switch can be up to 100Mbps.

The programs for assembly editing are uploaded into each assembly edit station by VTR or FC of linking other departments of a TV station. Each assembly edit station edits and links programme compressed with M-JPEG format, then it stores them into FC hard disk array.

The complete programme assembly edited are downloaded through FC download ports of each assembly edit station from FC hard disk array (as assembly edit database) to broadcast control. The programme tables amended are transmitted from FC server to broadcast control before program are to be downloaded. Then FC server sets the space of FC hard disk array free according to index tables of downloading program.

The assembly edit and download of programmes are parallel operations so that the broadcasting of 6 channels is continuous.

2.2 Broadcast Control

The heart of Broadcast Control consists of a 16 ports FC switch, a Video Server, a Files Server, a VOD edit Server, a Broadcast Control Server (BCS), a Transmit Stream (TS) Multiplexer, a 64 QAM modulation set, a RF modulation and mixer.

By a 16 ports FC switch the Video Server connects with download FC port of each assembly edit station, VOD edit Server, (BCS) and Files Server of connecting with Ethernet, Internet and the line of CATV via a Cable Modem.

The FC network of Broadcast Control is a video server oriented FC network. The Video Server also connects With a mass storage of FC hard disk array via a FC HUB as a video database (VDB), but all of the input Video programmes broadcasted must be pre-processed by a video server, before being stored into VDB.

The pre-processing of video server includes two parts: the first is decoding of programmes compressed in terms of M-JPEG format, then all broadcasted video programmes must be encode-compressed in terms of MPEG-2 format (MP@ML or MP@HL for HDTV)(it has higher compression ratio for reducing code stream rate and transmittal bandwidth). Above decoding and encoding can be performed in high speed by the decode/encode hardware card plugged into Video Server. The second part is striping programmes and creating distribution tables (6 broadcast channels and 2 VOD channels).

The I/O rate of FC hard disk array as VDB is over 800Mbps, so each disk can be in parallel accessed to satisfy needs of input and output in high speed of the video streams of 8 channels. If the compression ratio is 27:1(MPEG-2 MP@ML) and the rate of code stream is 6Mbps, the storage of a FC hard disk array as VDB may be chosen; 180G×4 to store video programmes for 64×2.5 hours (or it is used to store programmes of MPEG-2MP@HL format 15Mbps video stream and replay of VOD for 64 hours). The FC network of Broadcast Control is also managed with FC Server.

In the video server, there is a plug in Data Stream Adapter (DSA), including code stream distribution and driver software. The function of DSA is to pack digital code streams of programmes into serial frames of 188 bytes as indicated by a programme table, then to distribute in rate of 200Mbps digital code streams of programmes. For reducing the burden of CPU, the DSA uses DMA to read data of video streams from main RAM through X10-PCI Bridge. The programmes to have been broadcasted are immediately cancelled by video server so that the video server sets space of FC hard disk array free

Because video servers must process multi-channel digital video code streams of programmes in parallel, the host of a video server must be a computer with high performance (for example SGI Origing 2000 Rack (16pu)).

The VOD system between a TV station and a group of terminal users is a architecture with several layers (Ref.1), resulting in the number of channels occupied by VOD of a TV station being limited.

VOD edit servers performs the management of VOD service, CA and managing users, so in VOD edit server corresponding software are installed. For assembly editing programmes of VOD the hard disk of this Dayang 3000 LE as VOD edit server is changed (to be increased storage of hard disk or to be connected with Ultra SCSI-2 disk array (36G*2)).

The VOD edit server creates programme table according to the users' requests, then uploads video programmes via DVD, VTR or FC line connecting with Programme Database (PDB) of TV Station for assembly editing programmes. The download of programmes of VOD edit server is the same as assembly edit station.

VOD edit server process the ID and Key of each user. The user's ID is recorded and encrypted by VOD edit server when the user registers in VOD system via a telephone line. The user's Key is created (or updated) according to the user's register and recorded by the VOD edit server, then it is sent to Broadcast Control Server (BCS). BCS sends the key to TS Multiplexer (TSM). On one hand the key is used to interfere corresponding video data stream of VOD by TSM. On the other hand, it is inserted into SI data stream by TSM and will be sent to the terminal user for un-interference of VOD programme interfered.

The table of payments for ,VOD service of users is stored in the VOD edit server, so VOD edit server can interrupt or restart VOD service for one user according to the record of user's payment.

The Files Server (FS) provides service of public information received from Ethernet and Internet. The FS assembly edits public information and transforms them from common format into DVB-SI format, and then it sends SI data stream to Video Server. As TS stream SI data stream is send to TSM by Video Server.

The FS is also connected with line of CATV by a cable Modem for satisfying the need of users for Wide Band Telenet Service (WBTS)(it is being developed in Wuhan). Of course to use cable Modem must occupy some channels of CATV, so it is necessary to improve the line of CATV.

The Digital Satellite TV is TS of MPEG-2 format, thus it can be directly sent to TSM.

The BCS gets tables of programmes (including VOD) from Video Server, and then it sends various kinds of control commands according to tables of programmes and status of Broadcast system for monitoring broadcast.

In Fig.1 the number of TSM, 64QAM and RF modulation set are 4 groups. Each TSM has 4 input ports of MPEG-2 TS and the rate of code stream of each port is 0-15Mbps, sot it is enough to transmit programmes of 15 channels of Digital TV (including rediffusion of Digital Satellite signal of TV) by 4 channels of Analog CATV. If the numbers of rediffusion (or Self-broadcast and VOD Service) need to be increased, the number of TSM, 64QAM and RF modulation set must be increased.

The more detailed parts of hardware and software of each part are omitted. This plan design is based on present Analog CATV Telenet; it is also applicable for Digital CATV Telenet in the future.

REFERENCES

[1] Zhang Bixiong. "Analysis on System of Digital Television and VOD Based on Cable Television Network", Computer and Communications, 2001 (1)

[2] Zhu Tao. "The Digital Broadcast System of Shanghai CATV Station", from http://www.dicit.com (or http://www.myhome.tv)

[3] Y. Paker. "Custom TV Systems/Demonstration and Specification", from http://www.irt.de/customtv/

Segmentation of Tongue Image Using Color Edge Detector and Boundary Tracing Technique

Liu Guansong
College of Information and Engineering, East China University of Science and Technology
Shanghai, 200237, P. R. China
Email: maillgs@263.net
Lv Jiawen
Engineering Design and Research Institute, East China University of Science and Technology
Shanghai, 200237, P. R. China
Xu Jianguo
School of Basic Medical Sciences, Shanghai University of Traditional Chinese Medicine
Shanghai, 200032, P. R. China
Gao Dunyue
College of Information and Engineering, East China University of Science and Technology
Shanghai, 200237, P. R. China

ABSTRACT

In this paper, a segmentation algorithm for tongue images based on the color edge detector and the boundary tracing technique is presented. The color edge detector is obtained after reconstructing the grayscale Sobel operator. It is based on the color difference and can detect the edges of color images in parallel. After edges of tongue images are extracted, the tongue boundary line can be detected using the inner boundary tracing technique. Experimental results show the effectiveness of the segmentation algorithm.

Keywords: Image Segmentation, Tongue Image, Color Edge Detection, Sobel Operator, Boundary Tracing

1. INTRODUCTION

The principles of traditional Chinese medical diagnosis are based on the information obtained from four diagnostic processes, which are inspection, listening and smelling, inquiry and palpation. Among these diagnostic processes, the examination of tongue is one of the most important approaches for getting significant evidences in diagnosing the patient's health conditions. However, the clinical competence of tongue diagnosis was determined by the experience and knowledge of the physicians who adopted the tongue diagnosis. Most of the precious experiences in traditional tongue diagnosis could not be retained scientifically and quantitatively. Therefore, it is necessary to build an objective diagnostic standard for tongue diagnosis. The effectual measure is to design a computerized tongue examination system based on some image processing techniques for the purpose of providing a systematic and quantitative diagnostic standard and lowering the diagnosis among physicians [1]. It is of an important step to segment a tongue from its background view of the face in computer analysis of tongue image, and the veracity of subsequent algorithm for tongue image analysis depends directly on the segmentation results.

One of the most important image features is the 'edge' or the boundary line of a uniform region. Edges are useful clues to ways of segmenting an image into meaningful regions. The edge is defined as a series of edge points where the pixel value (intensity, color or range) change abruptly [2]. One way to extract edges from an image is called an 'edge detection'. Edge detection turns out to be a powerful tool for image processing and has received considerable attention in recent years. But, most of the methods of edge detection concentrate on grayscale images, and few of them on the color images. Since color images provide more information than grayscale images, more detailed edge information is expected from color edge detection. Most color edge detection schemes are based on finding maximal in the gradient of the image function or zero-crossings in the second derivation of the image function [3-6].

This paper presents a color edge detector developed from Sobel operator for the edge detection in color images. The detector is based on the CIE La*b* uniform chromatic space, which is intimately related to the way with which human beings perceive color. The detection can be executed in parallel, because the processing of a point of the image depends only on the local neighbor and does not depend on the results of processing for other points. In the paper, the color edge detector and the boundary tracing technique are combined successfully to segment meaningful tongue region from the color tongue image.

2. TONGUE IMAGE SEGMENTATION ALGORITHM

Color Edge Detector

For a grayscale image, edges are typically modeled as brightness discontinuities. That is to say, edges reflect the changes of gray value in grayscale image. Since the changes can be measured with the gradient of image function, the edge detector can be worked out with the local derivative technique [7].

The gradient $\nabla f(x,y)$ of the image $f(x,y)$ is defined as follows:

$$\nabla f(x,y) = [\partial f / \partial x, \partial f / \partial y]^T = [G_x, G_y]^T \quad (1)$$

Then, the magnitude of $\nabla f(x,y)$ is denoted as:

$$G(x,y) = \sqrt{G_x^2 + G_y^2} \quad (2)$$

The local direction of edge is provided by:

$$\Phi(x,y)=\arctan(G_x/G_y) \quad (3)$$

The derivatives of the three equations above must be computed to every pixel. In actual application, they are obtained approximately by convolving the image with local region templates. A gradient operator consists of two templates as G_x and G_y has individual templates respectively. Many gradient operators have been proposed, one of them is Sobel operator (see Fig. 1), which is applied widely to detect the edges of grayscale images.

In color images, edges are the regions that involve abrupt changes of not only brightness but also hue. The changes can

−1	0	1
−2	0	2
−1	0	1

1	2	1
0	0	0
−1	−2	−1

Fig. 1 Sobel operator

be defined as color differences. Since the description of color differences between two colors must be uniform to human being's vision, the CIE La*b* is chosen as the color space in the paper. In La*b* color space, L is the luminance, a* and b* are respectively red/green and yellow/blue chrominances. La*b* color space is approximately uniform for the perception of small color difference. That is, for specimens compared to standard, color differences in any direction are of bout the same importance (weight). Thus, the color difference is an equally weighted combination of the coordinate differences. Let ΔC denotes the color difference between the color $C_1=(L_1,a_1^*,b_1^*)$ and the color $C_2=(L_2,a_2^*,b_2^*)$, then ΔC is typically expressed as:

$$\Delta C=\sqrt{(L_1-L_2)^2+(a_1^*-a_2^*)^2+(b_1^*-b_2^*)^2} \quad (4)$$

The techniques of edge detection in grayscale images can be extended to color images. Look at the Sobel operator described above more closely; it is actually applied to compute the sum of the grayscale difference between the right pixels and the left pixels (or the top pixels and the bottom pixels). Since the grayscale difference is replaced by the color difference in color images, the grayscale gradient operator can be extended to color images:

Define the color difference between the pixel (x_1,y_1) and the pixel (x_2,y_2) as $\Delta C(x_1,y_1;x_2,y_2)$, from Eq. (4) we can obtain:

$$\Delta C(x_1,y_1;x_2,y_2)=\sqrt{[L(x_1,y_1)-L(x_2,y_2)]^2+[a^*(x_1,y_1)-a^*(x_2,y_2)]^2+[b^*(x_1,y_1)-b^*(x_2,y_2)]^2} \quad (5)$$

So, the color edge detector based on the grayscale Sobel operator is produced:

$$G(x,y)=\sqrt{\begin{matrix}[\Delta C(x+1,y-1;x-1,y-1)+2\Delta C(x+1,y;x-1,y)+\Delta C(x+1,y+1;x-1,y+1)]^2+\\ [\Delta C(x-1,y-1;x-1,y+1)+2\Delta C(x,y-1;x,y+1)+\Delta C(x+1,y-1;x+1,y+1)]^2\end{matrix}} \quad (6)$$

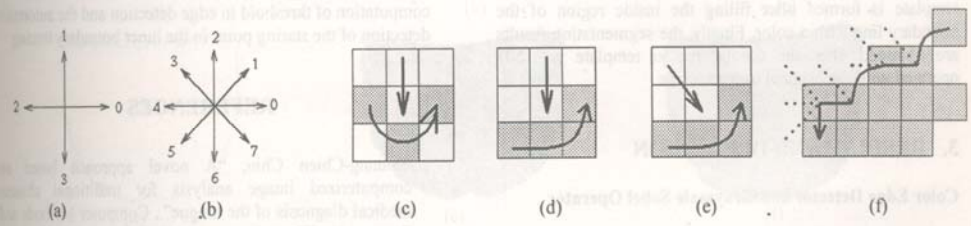


Fig. 2 Inner boundary tracing: (a) Direction notation, 4-connectivity, (b) 8-connectivity, (c) pixel neighborhood search sequence in 4-connectivity, (d), (e) search sequence in 8-connectivity, (f) boundary tracing in 8-connectivity (dashed lines show pixels tested during the border tracing)

Edge detection consists of three steps: first, convert the RGB color image into the La*b* representation; second, the gradient image (scaled to the 0~255 range) is obtained by using the color edge detector to the La*b* image; and finally, edges were extracted after thresholding.

Inner Boundary Tracing

Boundaries are linked edges that characterize the shape of an object. They are useful in computation of geometry features such as size or orientation. Conceptually, boundaries can be found by tracing the connected edges. On a rectangular grid a pixel is said to be four-or eight-connected when it has the same properties as one of its nearest four or eight neighbors, respectively. The algorithm for tracing closed inner boundaries in binary image is given:

1) Let pixel P_0 is a starting pixel of the region border. Define a variable *D, which* stores the direction of the previous move along the border from the previous border element to the current border element. Assign: (a). D =3 if the border is detected in 4-connectivity (Fig. 2(a)); (b). D =7 if the border is detected in 8-connectivity (Fig. 2(b)).

2) Search the 3×3 neighborhood of the current pixel in an

anti-clockwise direction, beginning the neighborhood search in the pixel positioned in the direction: (a). $(D+3)$ *mod* 4 (Fig.2(c)); (b). $(D+7)$ *mod* 8 if D is even (Fig.2(d)), or $(D+6)$ *mod* 8 if D is odd (Fig.2 (e)). The first pixel found with the same value as the current pixel is a new boundary element P_n. Update the D value.

3). If the current boundary element P_n is equal to the second border element P_1, and if the previous border element P_{n-1} is equal to P_0, stop. Otherwise repeat step (2).

4). The detected inner border is represented by pixels P_0 --- P_{n-2}.

Tongue Image Segmentation Procedure

The procedure of tongue image segmentation is shown in Fig. 3. First, the color edge detector described above is applied to the original color image to produce the gradient magnitude map. Second, the gradient magnitude map is converted into the edge map after thresholding. Third, the closed tongue boundary line is obtained by the inner boundary tracing technique. Fourth, the tongue region template is formed after filling the inside region of the boundary line with a color. Finally, the segmentation results are obtained after the tongue region template is AND operated with the original tongue image.

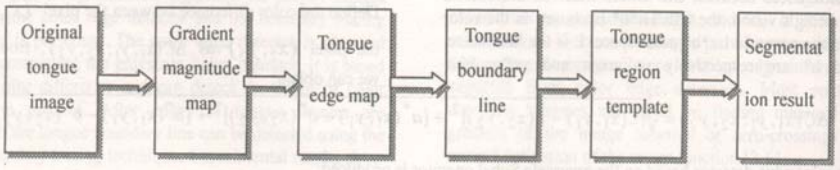


Fig. 3 The diagram of the tongue segmentation procedure

3. RESULTS AND DISCUSSION

Color Edge Detector and Grayscale Sobel Operator

To compare the color edge detector presented with the grayscale Sobel operator, the RGB color image is converted into grayscale image, then edges are detected with grayscale Sobel operators. In order to compare the results under the same condition, the threshold is chosen so that the number of edge pixels is in the equal percentage of the total number of pixels. Fig. 4 (13% of the total number of pixels) and Fig. 5 (18% of the total number of pixels) show the original color images and their detected results with color edge detector and grayscale Sobel operator. It is easy to find that the color edge detector can detect edges of the color image more correctly. For example, the "greenery" of the bouquet in the color image "GIRL" (see Fig. 4) is not extracted by the grayscale Sobel operator while is clearly detected by the color edge detector. The reason is that the color edge detector uses not only the brightness information but also the hue information

Tongue Image Segmentation

The tongue image segmentation algorithm presented was implemented and tested on tongue images. Fig. 6 gives the results produced in different segmentation stages, and Fig. 7 shows two other tongue images and their segmentation results. It is clearly shown that the algorithm proposed provides accurate results in the segmentation of tongue image.

4. CONCLUSIONS

In this paper we have described an algorithm for the segmentation of tongue image that uses the color edge detector and boundary tracing technique. The color edge detector is obtained after reconstructing the grayscale Sobel operator. It is based on the color difference and can detect the edges of color images in parallel. Experiments demonstrate that the proposed algorithm provides ideal results in the segmentation of tongue images.

Future work on this algorithm should include the automatic computation of threshold in edge detection and the automatic detection of the staring point in the inner boundary tracing.

REFERENCES

[1]Chuang-Chien Chiu, "A novel approach based on computerized image analysis for traditional chinese medical diagnosis of the tongue"., Computer Methods and Programs in Biomedicine ,vol. 61, 2000, pp. 77~89.

[2]Yoshiaki Shirai, Three-Dimensional Computer Vision, Springer-Verlag, Berlin, 1987.

[3]G. S. Robinson, Color edge detector, Optical Engineering, vol. 16, 1977, pp. 479~484.

[4]M. Hedley and H. Yan, Segmentation of color images using spatial and color space information, Journal of Electronic Imaging, vol. 1, 1992, pp. 374~380.

[5]T. Carron and P. Lambert, Color edge detector using jointly hue, saturation and intensity, in Proc. IEEE International Conference on Image Processing, 1994, pp. 977~981.

[6]A. Cumani, Edge detection in multispectral images, Graphical Models and Image Processing, vol. 53, 1991, pp. 40~51.

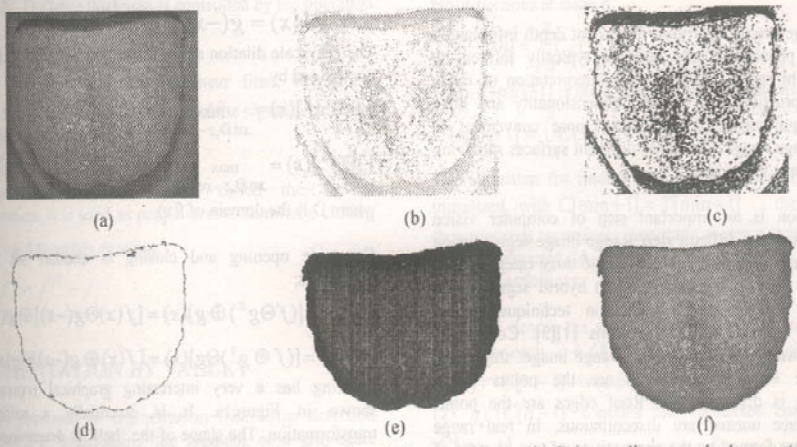
[7]Cui Yi, Techniques and Applications of Digital Image Processing, Publishing House of Electronics Industry, Bei Jing, 1997.

(a) (b) (c)

Fig. 4: Color image "GIRL" and detected results: (a) original image ; (b) with grayscale Sobel operator; (c) with color Sobel operator.

(a) (b) (c)

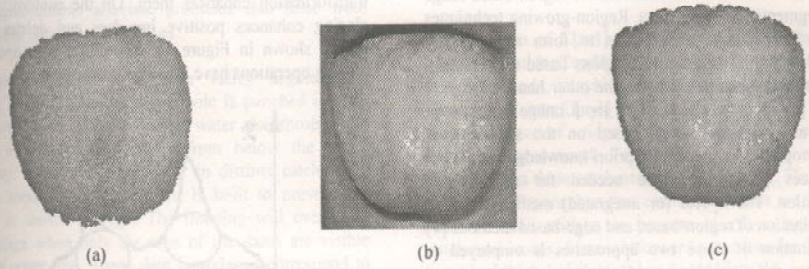
Fig. 5: Color tongue image and detected results: (a) original image ; (b) with grayscale Sobel operator; (c) with color Sobel operator.



(a) (b) (c)

(d) (e) (f)

Fig. 6 The results in different segmentation stages: (a) Original tongue image; (b) Color gradient magnitude map; (c) Thresholded tongue map; (d) Tongue boundary line; (e) Tongue region template; (f) Segmented tongue.



(a) (b) (c)

Fig. 7 Segmentation results of tongue images: (a) Result of Fig. 5(a); (b) Original tongue image; (c) Result of tongue image (b).

Segmentation of Range Image Based on Mathematical Morphology

Tao Hongjiu
Wuhan University of Technology
Wuhan, Hubei, 430063, China
Email: thjll@263.net taohongjiu@sohu.com

ABSTRACT

This paper presented a method of range image segmentation based on Mathematical Morphological. At first, morphological edge detector and threshold edge map extract the edges, then valley segmentation method is introduced to finish the last segmentation. The experimental results show this method has the better performance.

Keywords: Range Image, Segmentation, Valley, Segmentation

1. INTRODUCE

Range images carry viewpoint dependent depth information about the physical scenes and are typically formed by time-of-flight range-finders. Direct interpretation of range data is impractical due to high dimensionality and huge storage requirements. It is instead more convenient to segment range image points into different surfaces satisfying some similarity constraint.

Segmentation is an important step of computer vision system, it is also a difficult step. Range image segmentation techniques can be broadly classified into three categories: (i) edge-based (ii) region-based and (iii) hybrid segmentation techniques. Edge-based segmentation techniques detect boundaries between different regions [1][2]. Commonly, There are two types of edges in a range image: step edges and crease edges. Step edges are the points where range-value is discontinuous. Roof edges are the points where surface normal are discontinuous. In real range images, edges formed by the composition of two or more of these primitive edges are also present. In general edge-based methods suffer from broken edges contours and spurious edge points. There are two main classes region-based range image segmentation techniques. Region-growing techniques obtain a connected set of pixels to form a region by repeatedly merging neighboring regions based on similarity of the surface properties [3]. On the other hand, clustering methods partition the pixels of an input image into several clusters of connected pixels based on the similarity of surface properties. In general, a priori knowledge of number of surfaces present may be needed for region-based segmentation. The hybrid (or integrated) method refers to the combination of region-based and edge-based methods [4]. A combination of these two approaches is employed to overcome the problems of oversegmentation and undersegmentation, which are generally encountered in the edge-based and region-based methods.

In this paper, we develop a segmentation method of range image based morphology. At first, we use morphology edge detectors to extract the edge map, which do not lend themselves to traditional thresholding techniques to produce a final segmentation result. We show that using edge map as the input to a morphological valley segmentation segmentation algorithm yields rugged and consistent results. The experiment shows the good result.

2. EDGE DETECTOR OF MORPHOLOGY

The tools for grayscale morphological operations are simple functions g(x) having domain G. They are called structuring functions. Their symmetric counterparts are given by:

$$g^s(x) = g(-x)$$

The grayscale dilation and erosion of a function f(x) by g(x) are defined by

$$[f \oplus g^s](x) = \max_{x \in D, z-x \in G} \{f(z) + g(z-x)\} \quad (1)$$

$$[f \Theta g^s](x) = \max_{x \in D, z-x \in G} \{f(z) - g(z-x)\} \quad (2)$$

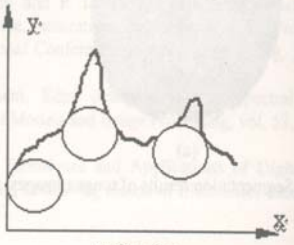
where D is the domain of f(x).

Grayscale opening and closing is another set of dual operations:

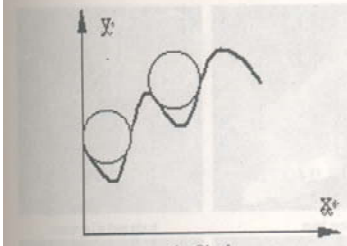
$$f_g(x) = [(f \Theta g^s) \oplus g](x) = [f(x) \Theta g(-x)] \oplus g(x) \quad (3)$$

$$f^g(x) = [(f \oplus g^s) \Theta g](x) = [f(x) \oplus g(-x)] \Theta g(x) \quad (4)$$

Opening has a very interesting graphical representation shown in Figure1a. It is essentially a rolling ball transformation. The shape of the 'ball' is determined by the structuring function g(x). The rolling ball traces the smooth contours and deletes the protruding (positive) impulses. When negative impulses are encountered, rolling ball transformation enhances them. On the contrary, grayscale closing enhances positive impulses and deletes negative ones , shown in Figure 1b. Therefore, both opening and closing operations have low-pass characteristics.



a: Opening



b: Closing

Figure1: Opening and Closing as a rolling ball transformation

Morphological operators can be used as edge detectors. The erosion $[f \Theta B](x)$ tends to decrease the area of high-intensity image plateaus. Therefore, define edge detector as follows:

$$EGf(x) = f(x) - [f \Theta g](x) \quad (5)$$

$$DGf(x) = [f \oplus g](x) - f(x) \quad (6)$$

EG[equation (5)] and DG[equation (6)] are erosion gradient and dilation gradient, so the edge detector is

$$ESf = \min[EGf, DGf] \quad (7)$$

The shape of structuring element g determines the edge orientation. The edge thickness is controlled by the operation times.

Opening f_B is a low-pass nonlinear filter, because it destroys the high- frequency content of the signal. Therefore, the algebraic difference

$$P(f) = f(x) - f_B(x) \quad (8)$$

is a nonlinear high-pass filter, called the top-hat transformation, it is used as peak detector. Similarly, closing f^B can used to valley detector

$$V(f) = f^B - f \quad (9)$$

3. SEGMENTATION BY VALLEY

A classic approach to segmentation from edge detection consists of the thresholding the gradient of a gray-scale image to produce a binary image: edge map. This particular approach is plagued by a number of practical limitations; such as the need for linking broken use of morphological valley segmentations alleviate problems that arise from classical edge detection techniques.

A conceptual description of the valley segmentation algorithm follows. Suppose that a hole is punched in each regional minimum and that letting water rise through the holes at a uniform rate floods from below the entire topography. When the rising water in distinct catchment's basins is about to merge, a dam is built to prevent the merging, as shown in Fig1, The flooding will eventually reach a stage when only the tops of the dams are visible above the water line. These dam boundaries correspond to the divide lines of the valley segmentations and, therefore, represent the edges extracted by a valley segmentation algorithm.

The concept of a valley segmentation is based on three types of points (a) points belonging to a reginal minimum, (b) points at which a drop of water could fall equally to more than one minimum.

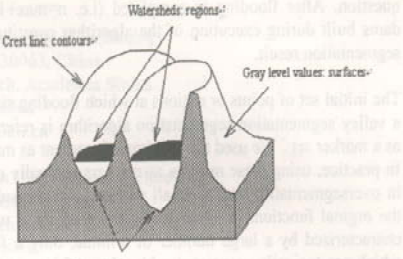


Figure 2: Gray-tone Valley segmentation algorithm

For a particular regional minimum, the set of points satisfying condition (b) is termed teh catchment basin or wateshed of that mionimum. The points satisfying condition (c) form crest line on the topographic surface and are the desired segmentation edges in athe gradient image. The principal objective of this class of segmentation algorithms is to find the valley segmentation lines.

Let C[n] denote the union of the flooded catchment basin-portions at stage n:

$$C[n] = \bigcup_{i=1}^{R} C_n(M_i)$$

Then, C[max+1] is the union of all catchment basins:

$$C[\max+1] = \bigcup_{i=1}^{R} C(M_i)$$

The algorithm for finding the valley segmentation lines is initialized with $C[\min+1] = T[\min+1]$. the algorithm then proceeds recursively, assuming at step n that C[n-1] has been constructed. A procedure for obstaining C[n] from C[n-1] is as follows.

Let Q[n] denote the set of connected components in T[n]. Then , for each $q \in Q[n]$, there are three possibilities:

(1) $q \cap C[n-1]$ is empty

(2) $q \cap C[n-1]$ contains one connected component of $C[n-1]$

(3) $q \cap C[n-1]$ contains more than one connected component of $C[n-1]$

Constructinos of C[n-1] depends on which of these three conditions holds. Condition 1 occurs when a new minimum is encountered , in which case connected component q is incorporated into C[n-1] to form C[n]. Condition 2 occurs when all or part of a ridge separating two or more catchment basins is encountered. Further flooding would cause the water level in these catchment basins to merge. Thus, a dam (or dams, if more than two catchment basins are involved) must be built with q to prevent overflow between the catchment basins. A straightforward procedure for building a one-pixel-thick (skeleton) dam is to dilate the components in X , with the dilation being constrained inside q. A typical stucturing element used for this purpose is a 3*3 mask of 1s. During dilation, pixels are appended to connected components, as long as merging between connected components does not occur. The iteration stops when no

more pixels can be appended . Teh resulting gap between connected components is the thin dam. The pixels in this gap are then projected up to a value max +1 to establish a permanent separation (bounary)between the regions in question. After flooding is completed (i.e. n=max+1), the dams built during execution of the algorithm constitute the segmentation result.

The initial set of points or regions at which flooding starts in a valley segmentation segmentation algorithm is referred to as a marker set . We used the mimima of gradient as markers. In practice, using these minima as markers generally results in oversegmentation due to small variations in the value of the orginal function. in other words, the gradient is usually characterized by a large number of minima, only a few of which are typically associated with edges of interest. Thus execution of a valley segmentation algorithm is usually preceded by a preprocessing step desinged to provide meaningful markers to the procedure.

4 CALCULATION STEPS AND EXPERIMENT RESULT

the experiment range image is a part range data, the size is 131×200, the max range value is 255. We segment this image based the theory as last parts, the calculation steps are as follows:

a. Filtered by gaussian filter with mean 1, variance 1.0, window size 5*5, to alleviate the effects of noise;
b. Extract edge based equation (7) , and threshold to get binary edge map;
c. Find the regional minima of edge map, get the markers;
d. Distance transform and get the catchment's basins;
e. Extract crest line by valley segmentation.

Experiment result is shown in figure 3, where figure 3b is edge map with Log operator, figure 3c is the result based the method of this paper. It showed the method based morphology valley segmentation can get good edges, it clearly segment regions of range image, it also provides closed contours. The key of valley segmentation algorithm is to find correct markers and corresponding catchment's basin, However, it can result in dramatic oversegmentions when markers are not correct because of noise. Extraction of markers may not be in gradient map, it can be gotten by other methods.

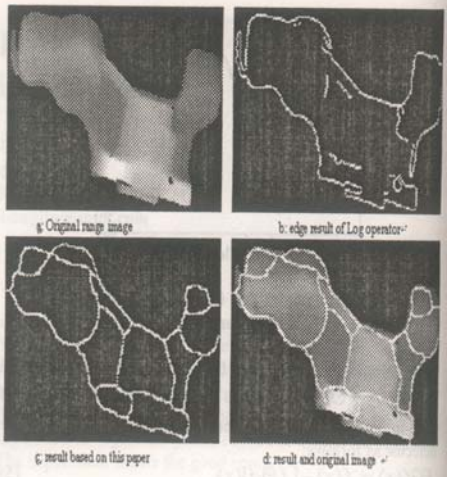


Figure 3: Experiment result

REFERENCES

[1] J.Berkmann and T.Caeli, Computation of Surface Geometry and Segmentation Using Covariance Techniques, IEEE Trans on PAMI, vol 16. No. 11, 1114-1116, 1994

[2] T. J. Fan, Describing and Recongnizing 3-D Objectes Using surface properties. Springer-Verlag, New York, 1990

[3] Besl and Jain, Segmentation Through Variable-Order Surface Fitting, IEEE Trans on PAMI vol 10, no.2 1988

[4] M.A. Wani and B. G. Batchelor, Edge-Region-Based Segmentation of Range Images, IEEE Trans on PAMI, vol. 16 no. 3, 1994

Point Match Algorithm Based on Alignment

Shengping Jin
Depart. Of Statistics, Wuhan University of Technology
Wuhan, Hubei Province, 430063, China
CAD Lab. of Inst. Computing Tech. Academia Sinica
Beijing, 100080,China
Email: spjin@public.wh.hb.cn
Dingfang Chen
Inst. of Intell. Manuf. & Control, Wuhan University of Technology
Wuhan, Hubei Province, 430063, China
CAD Lab. of Inst. Computing Tech. Academia Sinica
Beijing, 100080,China
Email: dfchen@public.wh.hb.cn

ABSTRACT

Points pattern matching is a popular approach when comparing two images, which may be obtained from different sensors, or from the same sensor at different times. The expenses for points pattern matching are very important in many applications. In this paper, we outline the general algorithm for the control point-matching problem. And an improved algorithm based on alignment is present. The algorithm is practicable and is used for the on-line verification of fingerprints as an example.

Keywords: Image, Point Matching, Fingerprint,Algorithm

1. INTRODUCTION

Comparing two images of the same scene is a task common to several different disciplines. The two images may be obtained from different sensors, or from the same sensor at different times. Applications arise, for example, in change detection (forestry, medicine, defense science), registration (satellite images, side-scan sonar pictures, maps, fingerprint) and model formation (stereo matching, topography).

When comparing images we have pursued a popular approach in that we have first looked for common points in both images. Since each pair of images usually contains millions of pixels, our approach has been to identify about a hundred feature points or minutiae in each image and then to match the common points (control points) in these subsets. Registration requires that both images coincide at the control points.

Thus, the problem can be stated as follows. Two sets of points in the plane are given. The first set contains m points. The second set is similar to the first set, except that some of the points from the first set are missing and some new points, not in the first set, are present. The second set contains n points: typically, m and n have values in the range 30-400. The positions of the points in the second set are, besides a translation or a rotation and within a tolerance, the same as common points in the first set.

The problem has three parts [1]:

(i) Find all points in the first set which do not have a match in the second set.
(ii) Find all points in the second set which do not have a match in the first set.
(iii) For all points in the first set which have a common point in the second set find the correct match: these correctly matched pairs are the control points.

2. ACCUMULATOR ALGORITHM

In D.Skea et al.[1] considered only translation in the second set and proposed an accumulator algorithm. An accumulator algorithm is a general scheme for matching planar points sets. The algorithm is robust in the presence of noise and can deal with missing and spurious points. Although the details of a particular implementation will vary depending on the type of positional errors present, the general procedure can be viewed in terms of three essential components. These are: (i) a geometric invariant property between points (or sets of points) by which trial matches can be evaluated, (ii) an accumulator where votes for a particular match are tallied, and (iii) a procedure for scanning the accumulator, once all trial matches have been made, to identify corresponding points.

The Geometric Invariant Property

The underlying assumption of any control point-matching algorithm is that there exists a mapping between the two sets of points, which can be characterized by some invariant property. If this property did not exist then a point in the first set could be matched with any point in the second set with an equal chance of it being the "correct" match. D.Skea et al.[1] used centroid and distortion of triangle to measure the matching points.

The Accumulator

The second component of the algorithm is the accumulator. An accumulator is an $m \times n$ array A in which votes for possible matches are tallied. Each of the m rows in the accumulator represents one of m points in epoch one and each of the m columns represents one of n points in epoch two. When the point j in epoch two is selected as a possible match with the points i in epoch one, then the value of the cell a_{ij} is incremented.

The Scanning Algorithm

The last component of the algorithm is a procedure to scan the accumulator and extract the correct matches. If the geometric

property selected if invariant with respect to the distortion, which is present in the image, then it would be expected that all correct matches would have large corresponding values in the accumulator. A possible scanning algorithm is as follows. Each row of A is first scanned for the largest element over a given tolerance, while all other entries are set to zero. This tolerance is set so that spurious points in one epoch can be detected as being missing in the other. Once each row has been scanned then each column is checked for multiple entries. In this case two or more points from epoch one have selected one point in epoch two as the most likely match. Again the largest value is chosen as the correct match. A similar algorithm can be designed to extract "new" points in either of the two epochs.

3. ALGORITHM BASED ON ALIGNMENT

D.P.Huttenlocher and S.Ullman[2] proposed a alignment algorithm to point matching problem. We improve on the method and give the algorithm in detail as following. The algorithm has two stages:

(i) Alignment stage: Two points from epoch one and epoch two are selected to consider matching points. Each point in epoch one is transferred to polar represent respect to the selected point from epoch one. Each point in epoch two is transferred to polar represent respect to the selected point from epoch two.
(ii) Matching stage: An elastic string matching algorithm is used to match the points in their polar coordinate system.

Alignment of Point Pattern

Denote the first set and the second set as[3]

$P = \{(x_1^P, y_1^P), (x_2^P, y_2^P), \cdots, (x_m^P, y_m^P)\}$,
$Q = \{(x_1^q, y_1^q), (x_2^q, y_2^q), \cdots, (x_n^q, y_n^q)\}$

respectively. For the rth point (x_r^P, y_r^P) in P, transform all point in P into polar represent with the polar origin (x_r^P, y_r^P) and the direction x-direction.

$$\begin{pmatrix} r_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \\ \tan^{-1}\left(\dfrac{y_i - y_r}{x_i - x_r}\right) + \Delta\theta \end{pmatrix} \quad (1)$$

where we omit the upper script p and the $\Delta\theta$ satisfies

$$0 \le \tan^{-1}\left(\frac{y_i - y_r}{x_i - x_r}\right) + \Delta\theta < 2\pi, \text{ i.e.}$$

$$\Delta\theta = \begin{cases} 0 & \Delta x > 0, \Delta y > 0 \\ \pi & \Delta x < 0, \Delta y > 0 \\ 2\pi & \Delta x > 0, \Delta y < 0 \end{cases}$$

$$\Delta x = x_i - x_r, \Delta y = y_i - y_r$$

Transform all point in Q into polar represent with the polar origin (x_j^q, y_j^q) and the direction x-direction as in P in the similar way.

Aligned Point Pattern Matching Algorithm

The aligned point pattern-matching algorithm is as follow:

(i) For every i($1 \le i \le m$) and every j($1 \le j \le n$), select (x_i^P, y_i^P) and (x_j^q, y_j^q) as a possible matching pairs and go to (ii). If all point pairs are considered, go to (iv).

(ii) For point (x_i^P, y_i^P) and (x_j^q, y_j^q), transform P and Q into polar systems according to(1) respectively. And resort them as the polar θ ascending. The polar represents of P and Q regard (x_i^P, y_i^P) and (x_j^q, y_j^q) as reference are as following:

$$P_i^s = \left\{(r_1^P, \theta_1^P), (r_1^P, \theta_1^P), \cdots, (r_m^P, \theta_m^P)\right\},$$
$$Q_j^s = \left\{(r_1^q, \theta_1^q), (r_1^q, \theta_1^q), \cdots, (r_n^q, \theta_n^q)\right\}$$

(iii) The matching score i_score[i][j] for (x_i^P, y_i^P) and (x_j^q, y_j^q) is computed as following:

For every (r_k^P, θ_k^P) ($1 \le k \le m$) and (r_t^q, θ_t^q) ($1 \le t \le n$), if $|r_k^P - r_t^q| < \varepsilon$, compute the difference of polar angle $\theta_k^P - \theta_t^q$ (let the difference be $\theta_k^P - \theta_t^q + 2\pi$ if $\theta_k^P - \theta_t^q < 0$). Let i_score[i][j] be the maximum number of the difference of polar angle within some tolerance. Then go to (i)

(iv) Find the maximum value of i_score[i][j]. The corresponding difference of polar angle, which has maximum number, is the rotation angle. Take the i,j as reference, for (r_k^P, θ_k^P) ($1 \le k \le m$) and (r_t^q, θ_t^q) ($1 \le t \le n$), if

$$|r_k^P - r_t^q| < \varepsilon(r_k^P) \quad (2)$$

and the difference of polar angle $\theta_k^P - \theta_t^q$ (let the difference be $\theta_k^P - \theta_t^q + 2.\pi$ if $\theta_k^P - \theta_t^q < 0$) is equal to the rotation angle are the matching points.

As to the elastic string-matching algorithm, we think of the tolerance $\varepsilon(r)$ in (2). The $\varepsilon(r)$ will be changed as r changes. In general, $\varepsilon(r)$ becomes large as r becomes large and is computed as in (3) and (4).

$$\varepsilon(r) = \begin{cases} r\_low & if\ r\_size < r\_low \\ r\_size & if\ r\_low \le r\_size \le r\_upper \\ r\_large & if\ r\_size > r\_upper \end{cases} \quad (3)$$

$$r\_size = \frac{r}{\alpha}. \quad (4)$$

where r_low, r_upper are the low bound and the upper bound value of $\varepsilon(r)$. α is a given constant.

4. IMPLEMTATION AND PARALLEL APPROACH

The algorithm based on alignment is effectual for arbitrary rotation center and translation. It is simple in theory, efficient in discrimination, and fast in speed. We have implemented the algorithm based on alignment to deal with specifically to the verification of fingerprints. To demonstrate the effectiveness of our algorithm, two examples are given. The first example is two sets. The first set contains 100 points which are generated randomly and range are from 0 to 1000. The second set is generated from the first set with a rotation and a translation. Each point has a random error up to 2 units. The second set is then resorted random. The matching process took 54s on a PII 400M. Computation tests show that the matching results are independent on the rotation center and translation distances and the order of the points in each set. It took 223s to match the common points of similar large scale using accumulator algorithm on an IBM/386 reported in [1].

The second example is two sets of minutiae extracted respectively from two fingerprints of a same finger. The first set contains 25 feature points. The second set contains 28 feature points. With a few modification of (2), 15 matching pairs can be found. The matching process took 0.01s on a PII 400M. So this matching algorithm can be used for the on-line verification of fingerprints.

As for large number points sets, the algorithm based on alignment can be considered to parallel processing approach as following. we take 2 computers as example to illustrate the approach. Let $i_0=0$, $i_1=[m/2]$, $i_2=m$, for the r'th computer (r=1or 2), computer the matching score separately:

(i) For every $i(i_{r-1}<i\leqslant i_r)$ and every $j(1\leqslant j\leqslant n)$, select (x_i^p,y_i^p) and (x_j^q,y_j^q) as a possible matching pairs and go to (ii). If all point pairs are considered, go to (iv).

(ii) For point (x_i^p,y_i^p) and (x_j^q,y_j^q), transform P and Q into polar systems according to(1) respectively. And resort them as the polar θ ascending. The polar represents of P and Q regard (x_i^p,y_i^p) and (x_j^q,y_j^q) as reference are as following:

$$P_i^s=\left\{(r_1^p,\theta_1^p),(r_1^p,\theta_1^p),\cdots,(r_m^p,\theta_m^p)\right\},$$
$$Q_j^s=\left\{(r_1^q,\theta_1^q),(r_1^q,\theta_1^q),\cdots,(r_n^q,\theta_n^q)\right\}$$

(iii) The matching score i_score[i][j] for (x_i^p,y_i^p) and (x_j^q,y_j^q) is computed as following:

For every $(r_k^p,\theta_k^p)\,(1\leqslant k\leqslant m)$ and $(r_t^q,\theta_t^q)\,(1\leqslant t\leqslant n)$, if $|r_k^p-r_t^q|<\varepsilon$, compute the difference of polar angle $\theta_k^p-\theta_t^q$ (let the difference be $\theta_k^p-\theta_t^q+2\pi$ if $\theta_k^p-\theta_t^q<0$). Let i_score[i][j] be the maximum number of the difference of polar angle within some tolerance. Then go to (i)

(iv) After two computer finished the score computation, find the maximum value of i_score[i][j]. The corresponding difference of polar angle, which has maximum number, is the rotation angle. Take the I , j as reference, for $(r_k^p,\theta_k^p)\,(1\leqslant k\leqslant m)$ and $(r_t^q,\theta_t^q)\,(1\leqslant t\leqslant n)$, if $|r_k^p-r_t^q|<\varepsilon(r_k^p)$ and the difference of polar angle $\theta_k^p-\theta_t^q$ (let the difference be $\theta_k^p-\theta_t^q+2\pi$ if $\theta_k^p-\theta_t^q<0$) is equal to the rotation angle are the matching points.

REFERENCES

[1]D.Skea,I.Barrodale,R.Kuwahara and R.Poeckert. A Control Point Matching Algorithm. Pattern Recognition, V.26,N.2, 1993, pp.269-276.

[2]D.P.Huttenlocher and S.Ullman, Object recognition using alignment, Proc.ICCV,IEEE,1987

[3]A.Jain, Lin Hong and R.Bolle. On-Line Fingerprint Verification, IEEE Trans. Pattern Analysis and Machine Intelligence, V.19,N4, 1997,pp.302-313

[4]L.O'Gorman and J.V.Nickerson. An Approach to Fingerprint Filter Design, Pattern Recognition, V.22,N.1, 1989, pp.29-38

[5]A.Rosenfeld and A.C.Kak. Digital Picture Processing,V.2, 2nd Edition, Academic Press,1982.

A Distributed Algorithm for the Estimation of Heat Generation in a Welding Process

C.-H. Lai*, C.S. Ierotheous, C.J.Palansuriya, K.A. Pericleous
School of Computing and Mathematical Sciences, University of Greenwich
London SE10 9LS, UK

ABSTRACT

The inverse determination of heat generation due to a welding process is discussed. A mathematical model and its domain decomposition are presented. A distributed algorithm based on coarse-grained computing environment is also presented.*

Keywords: Inverse Problems, Welding, Parallel Algorithm.

1. INTRODUCTION

The welding of metals and alloys is a widely used industrial process. Many types of analysis have been carried out on such problems [8]. The numerical thermal analysis of welding is required to take into account such features as temperature dependent material properties, phase change, non-uniform distribution of energy from heat source etc. In this paper, a 2-D non-linear electric arc-welding problem is considered. It is assumed that the moving arc generates an unknown quantity of energy, which makes the problem an inverse problem with an unknown source. Automatic control of the time dependent source is of importance in the production of a fine weld. As such, robust algorithms to solve such problems and to retrieve the heat source efficiently, and in certain circumstances in real-time, are of great technological and industrial interest.

There are other types of inverse problems, which involve inverse determination of heat conductivity or material properties [3][10], inverse problems in material cutting [7], and retrieval of parameters containing discontinuities [5]. As in the metal cutting problem, the temperature of a very hot surface is required and it relies on the use of thermocouples. Here, the solution scheme requires temperature measurements lied in the neighborhood of the weld line in order to retrieve the unknown heat source. The size of this neighborhood is not considered in this paper, but rather a domain decomposition concept leading to a parallel algorithm is discussed. In addition, the mathematical model in this article, which involves heat conduction inside the material, is incorporated with a simple phase change model to handle any phase change.

This paper is organized as follows. The inverse problem is formulated and a method for the source retrieval is presented in Section 2. The source retrieval method is based on an extension of the 1-D source retrieval method as proposed in [6] for metal cutting problems. A parallel algorithm based on the concept of coupling heterogeneous numerical models in different subdomains is given in Section.

2. THE WELDING PROBLEM

* Corresponding author .C.H.Lai@gre.ac.uk

General Assumptions

Four assumptions are needed in this problem. These assumptions are (1) the material properties are homogeneous across the domain of interest, (2) application of a welding tool along a weld path is equivalent to the application of a heat source along the path, (3) the heat source at any point along the path is distributed over an infinitesimal neighborhood surrounding that point and, (4) the rate of change of temperature on either side of the weld is directly proportional to the strength of the heat source [6]. The welding problem considered in this paper is the welding of two thin metal plates using the technology of arc-welding.

Mathematical Model

For simplicity, the electric arc is moving along the weld path, y= y_w with a speed of u_w. Without loss of generality, the welding line can be a straight line or a general path. If the welding path was a straight line, it is always possible to align the welding path with the x-axis, i.e. the welding tool travels along y= y_w =0. The model can be further simplified to have the subdomains above and below the welding line being the same size. Therefore due to the symmetry of the problem only the upper half of the domain needs to be considered. The weld line used in the present analysis is depicted in Figure 1 as a dotted line which separates the computational domain into two equal parts. Since the thickness of the plate, d, is small compared to the other dimensions, only 2-D heat conduction needs to be considered. Hence, using the first two assumptions, the mathematical model, which governs the heat conduction of the plate, can be written as the following 2-D nonlinear, unsteady, parabolic, heat conduction equation,

$$c_e \frac{\partial T}{\partial t} \equiv \frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k(T)\frac{\partial T}{\partial y}) - 2h_{eff} A(T - T_a) + Q_w \quad (1)$$

subject to the initial condition T(x,y,0)= T_i (x,y) and boundary conditions defined by the functions:

$$B_0\,[T(0,y,t),0,y,t]=0$$

$$B_1\,[T(l,y,t),l,y,t]=0$$

$$C_0\,[T(x,-h,t),-h,t]=0$$

and

$$C_1\,[T(x,h,t),h,t]=0$$

Here T(x,y,t) is the temperature distribution, *k(T)* is the conductivity of the metal plates, t is the time, h_{eff} is the

effective heat transfer, A is the surface area, a T is the ambient temperature, $c_e = \rho c - L\frac{\partial f_1}{\partial T}$ is the effective specific heat, r is the density, c is the specific heat capacity, L is the latent heat, $\frac{\partial f_1}{\partial T}$ is the variation of liquid fraction, $Q_W = Q_W(x,y,t)$ is the time dependent heat source generated from the moving arc. T_1, B_0, B_1, C_0 and C_0 are known functions. The source term, Q_W in Eq. (1) is an unknown, and the inverse problem here is to retrieve this unknown heat source.

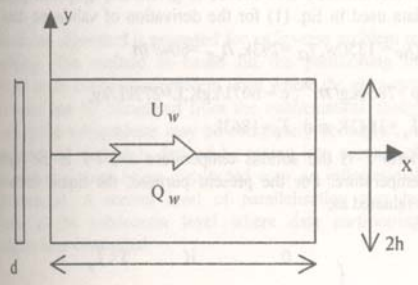


Figure 1: A simple welded work-piece.

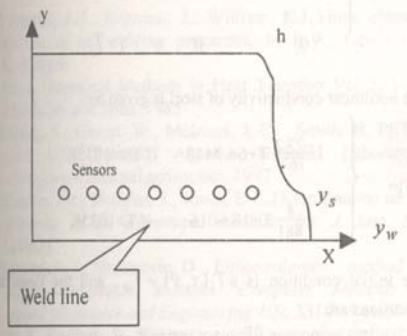


Figure 2: Locations of thermocouples.

The Inverse Problem

In order to deal with this additional unknown, temperature measurements near the weld line is required (see Figure 2). Thermocouples are attached along $y = y_s = y_w + \eta \min(l, h)$, such that $0 < \eta < 1$, Let the temperature measured by means of the thermocouples be $T(x, y_s, t) = T^n(x,t)$ The measured temperatures are used as interior boundary conditions, as Section 3, along subdomain interfaces and to retrieve the temperature distribution at the welding points. The heat source retrieval is based on the fourth assumption, i.e. in the neighborhood of the weld,

$$\frac{\partial T}{\partial t} = a(x,t)\, Q_W(x,t) \tag{2}$$

where $a > 1$ is a time dependent function that also depends on the spatial dimension x. The condition $a > 1$ is to ensure an increase in temperature at the weld due to an increase in Q_W Integrating Eq. (2) across the weld at a given value of x gives

$$\int_{y_w^-}^{y_w^+} \frac{\partial T}{\partial t} = a(x,t)\, Q_w(x,t)(y_w^+ - y_w^-) \tag{3}$$

Where y_w^+ *to* y_w^- *is* the width of the weld along y-axis at a given instance of time under immediate influence of the electric arc. Integrating Eq. (1) across the weld and equating the result to Eq. (3) lead to

$$\begin{aligned} & k(T)\frac{\partial T}{\partial y}|_{y_w^+} - k(T)\frac{\partial T}{\partial y}|_{y_w^-} \\ & + \frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x})(y_W^+ - y_W^-) \\ & - 2h_{eff}\, A(T - T_a)(y_W^+ - y_W^-) \\ & = (c_e a(x,t) - 1) Q_W(x,t) \end{aligned} \tag{4}$$

Let $\beta(x,y) = c_e\, a(x,y) - 1$ and define the predicted heat source as $Q_p = \beta(x,t)\, Q_W(x,t)$ which may be computed as

$$\begin{aligned} Q_p(x,t) &= k(T)\frac{\partial T}{\partial x}|_{y_w^+} - k(T)\frac{\partial T}{\partial y}|_{y_w^-} \\ &+ \frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x})(y_W^+ - y_W^-) \\ &- 2h_{eff}\, A(T - T_a)(y_W^+ - y_W^-) \end{aligned} \tag{5}$$

Then Q_p can be substituted into Eq. (1) to replace Q_W and the non-linear heat conduction problem may then be solved as a direct problem with $T_p(x,t)$ being the corresponding temperature distribution. Hence it is possible to evaluate $\beta(x,t)$ from the knowledge of $T_p(x,t)$ and $T(X, y_w, t)$ as

$$\beta(x,t) = \frac{T_p(x,y)}{T(x,y_w,t)} = \frac{Q_p(x,t)}{Q_w(x,t)} \tag{6}$$

Where $T(x, y_w, t)$ is the temperature at the weld line corresponds to $Q_W(x,t)$. Hence $Q_W(x,t)$ may then be determined once $\beta(x,t)$ is known. Note that it is not necessary to compute $c_e\alpha(x,t) - 1$.

3. THE PARALLEL ALGORITHM

Since the heat source is an extra unknown in the mathematical model, it makes sense to eliminate the unknown source term of the model [7] for the governing equations on both sides of the welding path. The monitored thermocouple data provides an ideal interior partitioning. For the present study, y_w is chosen as zero and the metallic plates above and below the welding line is of the same size. Hence the problem become symmetric and only half of the entire problem needs to be considered. The computational domain is partitioned to two well defined, homogeneous, continuous and properly connected subdomains denoted by

D_1 ={(x,y): 0<x<1 and 0<y< y_s }

D_2 ={(x,y) 0<x<1 and y_s <y<h}

and they are depicted as in Figure 3. The two subproblems can be written as follows:

sp_1:

$$c_e\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x}+\frac{\partial T}{\partial y}(k(T)\frac{\partial T}{\partial y})$$
$$-2h_{eff}A(T-T_a)\in D1$$

sp_2 :

$$c_e\frac{\partial T}{\partial t}=\frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x}+\frac{\partial T}{\partial y}(k(T)\frac{\partial T}{\partial y})$$
$$-2h_{eff}A(T-T_a)\in D2$$

Subject to suitable initial conditions the temperature are defined on two different subdomains of different sizes, which are subjected to different set of boundary conditions. They are non-linear in nature and are completely decoupled from each other. Therefore using the Newton's method may solve them simultaneously or concurrently. Let *F(T)* be defined as

$$c_e\frac{\partial T}{\partial t}-\frac{\partial}{\partial x}(k(T)\frac{\partial T}{\partial x}+\frac{\partial T}{\partial y}(k(T)\frac{\partial T}{\partial y})$$
$$-2h_{eff}A(T-T_a)=0$$

Which leads to the corresponding Jacobian *J(T)* as follows:

$$J(T)=$$
$$c_e\frac{\partial}{\partial t}-k'\frac{\partial^2 T}{\partial x^2}-k\frac{\partial}{\partial x^2}-k''(\frac{\partial T}{\partial x})^2-2k'\frac{\partial T}{\partial x}\frac{\partial}{\partial x}$$
$$-k'\frac{\partial^2 T}{\partial y^2}-k\frac{\partial^2}{\partial y^2}-k''(\frac{\partial t}{\partial y})^2-2k\frac{\partial t}{\partial y}\frac{\partial}{\partial x}$$
$$+2h_{eff}A(T-T_\alpha)$$

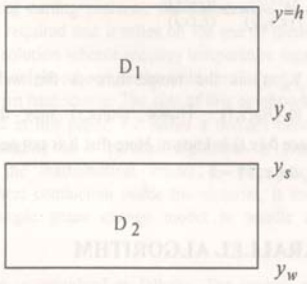


Figure 3: Subdomains and subproblems

The linearisation leads to an iterative scheme, to be performed in each of the subdomain,

$$T^{new}=T^{old}-J^{-1}(T^{old})F(T^{old})$$

Where superscript {new} denotes new iterates and {old} denotes old iterates. *F(T)* and *J(T)* are obtained by a second order finite volume method which leads to a set of large sparse linear system and it can be solved by means of a standard domain decomposition software package such as PETSc [2]. More processors may be used to achieve a secondary level of parallelism for the Newton's iterative scheme in each of the two subdomains, which are separately controlled by different hosts assigned to each of the subproblems. Therefore the inverse welding problem has two levels of parallelism. One level being the differential equation level and the other being the discretised level [7].

In this paper, parallelisation at the linearisation level is not discussed.

Validation

A validation problem for comparison purposes is defined here. Geometry of the two plates is chosen as l=0.5m, 2h=0.33m, d=0.008m .The true source is given in [1][9]. The physical data used in Eq. (1) for the derivation of validation data are

Q_w = 1350w, T_α =293k, h_{eff} =60w/m^2 ,

ρ =7850kg/m^3 , c = 607J/kgK,L=272kL/kg,

T_s =1843K and T_l =1863k.

Here T is the solidus temperature and T is the liquidus temperature. For the present purpose, the liquid fraction is evaluated as,

$$f_l=\begin{cases}0 & \text{if } T<T_s\\ \dfrac{T-T_s}{T_l-T_s} & \text{if } T_s\leq T\leq T_l\\ 0 & \text{if } T>T_l\end{cases}$$

The nonlinear conductivity of steel is given by:

$$k(T)=\begin{cases}\frac{-27.2}{762}T+64.9448 & \text{if } T<=1035K\\ \frac{8}{881}T+18.6016 & \text{if } T>1035K\end{cases}$$

The initial condition is a $T_i(x,y)=T_\alpha$, and the Boundary conditions are

$$B_0=B_1=k\frac{\partial T}{\partial x}+h_{eff}(T-T_\alpha)=0$$

and

$$C_1=k\frac{\partial T}{\partial v}+h_{eff}(T-T_\alpha)=0.$$

The source is traveled at the speed u_w = 0.00333m/s and applied only at cells which are at a given instant of time under immediate influence of the electric arc. Solving the direct problem with the given known source provides the temperature field of the steel plate. A mesh size of 50×50 is used to obtain the validation data. Thermocouple temperature measurements are available for comparison from MPA Stuttgart [1]. Detailed computation of validation data can be found in [9] which shows that the validation data matches with the experimental data. *h* is now chosen as 0.02 and, therefore the thermocouples are placed at y_s = 0.0033m suggested in [1]. Ideally, the temperatures recorded by the thermocouples are equivalent to the validation data obtained above at y_s = 0.0033m. The inverse problem given by Eq. (1) is solved by using the mesh configuration of 200×200 . The retrieved temperature distribution and the source over the welding path as compared in the thesis [9] are accurate.

Parallel Implementation

The approach proposed in this paper naturally gives a coarse-grained parallel algorithm. In other words, each subdomain generated can be mapped directly to a processor and the subproblems defined in each of the subdomains may then be solved concurrently. However such implementation obviously restricted in further performance enhancement. One way to increase the performance of the algorithm is to introduce data parallelism into each of the subdomains. This will enable the usual parallel algorithms being applied to each subdomain.

4. CONCLUSIONS

A distributed algorithm is proposed for an inverse problem in arcwelding. The method is based on the partitioning of problems at the continuous problem level where the unknown heat source can be eliminated from the mathematical model and where the subproblems may be completely decoupled. A source retrieval method is derived by taking into account of the source at a point being distributed over an infinitesimal neighbourhood. A second level of parallelisation may be obtained at the subdomain level where data partitioning techniques may be applied.

REFERENCES

[1] Argyris, J.H., Szimmat, J., William, K.J.,Finite element analysis of arc-welding processes. In R.W. Lewis and K.~Morgan (eds.), Numerical Methods in Heat Transferr Vol 3, 1-34. John Wiley and Sons, 1985

[2] Balay, S., Gropp, W., McInnes, L.C. Smith, B. PETSc 2.0 User Manual. Argonne National Laboratory, http://www.mcs.anl.gov/petsc/, 1997

[3]Cannon, J.R., Douglas, J., Jones, B.F.,Determination of the diffusivity in an anisotropic medium. *Int. J. Eng. Sci.* I,(1963)

[4]Demirdzic, I., Martinovic, D ., Finite volume method for thermo-elasto-plastic analysis. *Computer Methods in Applied Mechanics and Engineering 109*, 331-349 (1993)

[5]Ito, K., Kunisch, K., The augmented Lagrangian method for parameter estimation in elliptic systems. *SIAM J . Control Optim.* 28, 113-136 (1990)

[6]Ierotheou, C.S., Lai, C.-H., Palansuriya, C.J., Pericleous, k35.A., Espedal, M.S., Tai, X.-C.,Accuracy of a domain decomposition method for the recovering of discontinuous heat sources in metal sheet cutting. *Computing and Visualisation in Science* 2, 149-152 (1999)

[7] Ierotheou, C.S., Lai, C.-H., Palansuriya, C.J.,Pericleous, K.A., Simulation of 2-d metal cutting by means of a distributed algorithm. *The Computer Journal* 41, 57-63 (1998)

[8] Myers, P.S., Uyehara, O.A., Borman, G.L.,Fundamentals of heat flow in welding.*Welding Research Council Bulletin* 123,(1967)

[9] Palansuriya, C.J., Domain Decomposition Based Algorithms for Some Inverse Problems. PhD Thesis, University of Greenwich, 2000 .

[10] Tai, X.-C., Espedal, M.S.:, Rate of convergence of some space decomposition method for linear and non-linear elliptic problems *SIAM J. Numer. Anal*, 1558-1570 (1998)

Large-Scale Parallel Reservoir Simulation On Distributed Memory Systems

Cao Jianwen, Pan Feng, Sun Jiachang
R & D Center for Parallel Software, Institute of Software, CAS
P.O.Box 8718, Beijing 100080, P.R.China
Email: {cao,pan,sun}@mail.rdcps.ac.cn

Liu Wei
Baker Atlas GEOScience, Baker Hughes Inc
17015 Aldine Westfield, Houston, Texas 77073 USA
Email: wei.liu@bakerhughes.com

ABSTRACT

Computational efficiency and memory scalability are major subjects in large-scale reservoir simulations. In order to solve large-scale problems within required time, parallel codes running on distributed / shared memory systems are required.

This paper presents a parallel simulator developed especially for distributed memory systems, and some numerical results of test cases on parallel computers are given. Results show that this parallel code behaves satisfactory scalability and computational speedup.

The paper also presents Krylov subspace methods with two types of preconditioners, one based on ILU decomposition and the other based on an iterative algorithm. Numerical tests show that different Krylov subspace methods with an appropriate preconditioner are able to achieve similar performance.

Keywords: Large-Scale, Parallel Reservoir Simulation, Distributed Memory System, Preconditioner, Krylov Subspace Methods, Numerical Comparison.

1. INTRODUCTION

Petroleum reservoir simulation uses a numerical material balance approach to generate realistic development scenarios ([1-3]). Currently, many large-scale simulations are still limited by the CPU frequency and memory requirements. Moreover, reservoir description data often comes in finely gridded geostatistical models containing more grid cells than can be efficiently handled. The process of upscaling geostatistical grid cells to coarse grids becomes common practice, however, it inevitably introduce much uncertainties and inaccuracies in simulation results.

In the last few years, the performance of parallel reservoir simulation research has been significantly improved ([4], [6-7], [9-12]). At the mean time, complex geological environments and production management in large-scale reservoir simulation increasingly demands higher field resolution and larger computational resources.

The objective of our project is to develop a simulator that can be used for solving a variety of practical and challenging reservoir simulation problems. It is widely recognized that the most computationally expensive part is the solution of the sparse linear equations from finite-difference or finite-element discretizations. The widely used ILU preconditioning and its related techniques are sequential in nature and lead to poor efficiency of the implementation on distributed memory computer platforms. To overcome this problem, a domain decomposition method (DDM) based preconditioning technique and a set of hybrid preconditioners with Krylov subspace methods have been developed which allow for solving an important class of linear systems of interest in large-scale simulation applications. The technique provides significant speedup even when using a large numbers of processors, without sacrificing the resolution of solving problems.

In this project, the parallel simulator is portable to distributed / shared memory systems that support MPI (Message Passing Interface) for interprocessor communication. Efficiency, flexibility and portability were emphasized throughout the process of design and implementation.

A grid-partition strategy is used on all fine and coarse grids. The entire computational grid is partitioned and distributed to a logical network of processors. The solver package has been designed and coded so that it can be easily adapted to solving a variety of multicomponent, three-dimensional and three-phase flow problems, not being limited for black-oil type simulations.

2. SIMULATION MATHEMATICAL MODEL

Reservoir simulation solves the multidimensional and multiphase equations of conservation of mass in porous media, subject to appropriate initial and boundary conditions. The parallel version of the used simulator is limited to Black Oil Model currently.

Black Oil Model ([2]) is the most important part and is regarded as the fundament of reservoir simulation work. In this model there are three distinct phases namely, oil, water and gas. Fluids of different phases are usually considered to be at constant temperature and in thermodynamic equilibrium throughout the entire reservoir in consideration. The three-phase flow conservation equations can be expressed as:

$$-\nabla\cdot[T_w(\nabla P_w - \rho_w g\nabla D)] + q_w = \frac{\partial}{\partial t}(b_w S_w)$$

$$-\nabla\cdot[T_o(\nabla P_o - \rho_o g\nabla D)] + q_o = \frac{\partial}{\partial t}(b_o S_o)$$

$$\nabla\cdot[T_o R_s(\nabla P_o - \rho_o g\nabla D)] + \nabla\cdot[T_g(\nabla P_g - \rho_g g\nabla D)] + R_s q_o + q_g = \frac{\partial}{\partial t}(\phi\, b_o R_s S_o + \phi\, b_g S_g)$$

$T_l = M_l b_l$ $(l = w, o, g)$ is the transmissibility of phase-l.

As a factor of T_l, $M_l = \dfrac{KK_{rl}}{\mu_l}$ (the mobility) gives out a relationship between the flow rate and pressure gradient in each phase (Darcy's Law): $\vec{v}_l = -M_l(\nabla p_l - \rho_l g\nabla D)$

$$K = \begin{pmatrix} K_x(x,y,z) & & \\ & K_y(x,y,z) & \\ & & K_z(x,y,z) \end{pmatrix}$$

$K_{ro} = K_{ro}(S_w, S_g)$, $K_{rw} = K_{rw}(S_w)$, $K_{rg} = K_{rg}(S_g)$

$\rho_l = \rho_l(P_l)$, $b_l = b_l(P_l)$, $D = D(x,y,z)$

$S_w + S_o + S_g = 1$, $\phi = \phi(P_o)$, $R_s = R_s(P_o)$

$P_o - P_w = P_{cwo}(S_w)$, $P_g - P_o = P_{cgo}(S_g)$

$q_l = C(x,y,z)T_l(P_o - P_{wf})$

Some nomenclature in PDEs is described as bellow

The unknowns of above PDEs are oil-phase pressure (P_o), water-phase and gas-phase saturation (S_w, S_g). By means of considering special cases, we may know about their obscure characteristics. For single-phase case [1], [3], if some physical effects are neglected, or assumed to be constant, the following PDE are derived for two-phase (oil-water system) case [1]:

$\nabla^2 P = \dfrac{\phi\mu c}{K}\dfrac{\partial P}{\partial t}$ (liquid of slight compressib ility c)

$\nabla^2 P^2 \doteq \dfrac{\phi\mu c}{K}\dfrac{1}{P}\dfrac{\partial P}{\partial t}$ (ideal gas through porous media)

$\nabla^2(P - \rho g D) + \dfrac{\mu}{K\rho}\, q = 0$ (incompress ible flow)

by defining total velocity $\vec{v}_t = \vec{v}_o + \vec{v}_w$,

average pressure $P_{avg} = (P_o + P_w)/2$

and total compressibility

$$c_t = \frac{1}{\phi}\frac{d\phi}{dP_{avg}} + S_o c_o + S_w c_w;\; c_o \doteq \frac{1}{\rho_o}\frac{d\rho_o}{dP_o};\; c_w \doteq \frac{1}{\rho_w}\frac{d\rho_w}{dP_w},$$

Pressure and saturation equations can be expressed as

$$\nabla\cdot(M_o + M_w)\nabla P_{avg} \approx \phi c_t \frac{\partial P_{avg}}{\partial t}$$

$$\nabla\cdot h_w\nabla S_w - \frac{df_w}{dS_w}v_t\nabla S_w \approx \phi\frac{\partial S_w}{\partial t}$$

where $f_w \doteq \dfrac{M_w}{M_o + M_w}$, $h_w \doteq -f_w M_o \dfrac{dP_{cow}}{dS_w}$

First, the PDEs behave parabolic characteristics. Single-phase PDEs have the same form of heat conduction equations and maybe nonlinear. Two-phase PDEs superficially resemble heat conduction equations also. Second, the PDEs have the character of elliptic equations. The effects of compressibility usually don't dominate, especially for incompressible flow or slight compressible flow, thus, as a practical matter; the pressure equation must also be treated as being ecliptic or nearly ecliptic in nature. Third, the saturation equation can be regarded as a nonlinear variation of the diffusion-convection equation

$$D\nabla^2 C - \vec{v}\nabla C = \phi\frac{\partial C}{\partial t}$$

If the diffusion term dominates which means that h_w is large and capillary pressure effect dominates, this PDE behaves like a parabolic equation. However, if the capillary effects are small, when velocities are large, the convection term dominates, and this PDE will approach a first-order nonlinear hyperbolic equation. These characteristics require appropriate difference formulations and suitable preconditioned linear solvers in order to solve various applications efficiently. The details about the analysis of PDE characteristics can be found in [1] and [3].

3. NONLINEEAR SOLVER AND LINEEAR SOLVER

Finite difference formulation of the component conservation equation adopts block-centered grid system in the used simulator.

Considering convection-dominated PDEs, the choice of first-order difference scheme is crucial, both backward (upstreaming) and centered scheme in spatial direction can satisfy the requirement of unconditionally stable. Large timestep requirement discards the choice of forward-in-temporal scheme. Thus, there are four combinations of schemes available, backward-in-spatial and backward-in-temporal, backward-in-spatial and centered-in-temporal, centered-in-spatial and backward-in-temporal, centered-in-spatial and centered-in-temporal. All the four combinations may occur numerical dispersion or oscillation (overshoot) phenomenon. Numerical results and theory analysis assure that we can't avoid the two phenomena at the same time ([1], [3]). By choosing different combination, trade off between one and the other is available. In order to keep the scheme unconditional stables and avoids numerical solution oscillation, the backward-in-spatial and backward-in-temporal scheme is used in the process of discretization of the used simulator. In other word, we adopt fully implicit scheme in temporal and upsteaming scheme in special.

Fully implicit formulation leads to nonlinear difference equations; thus Newtonian iteration method is required. It is noted that nonlinearity of the model equation leads to timestep restriction also, though it is much less stringent than that for less-implicit difference scheme. General description of the nonlinear solver is as follows

Solving nonlinear equation $F(u) = 0$

Define $\delta u \doteq u^{n+1} - u^n$

(a) Give initial guess u^0

(b) For $n = 0,1,2,...$until convergence do

Using Taylor's formula, linearize nonlinear Eq.

$$F(u^{n+1}) = F(u^n + \delta u)$$

$$= F(u^n) + J(u^n)\delta u + O(\delta^2) \approx 0$$

solve the linear system and obtain δu

$$J(u^n)\delta u = -F(u^n)$$

update the vector $u^{n+1} = u^n + \delta u$ nonlinear convergence check When $F(u)$ is highly nonlinear, the above classical Newton method may not converge, its variants such as inexact Newton and preconditioned Newton iterations with trust region method may be used to instead. The linear system is expressed as

$$\begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}$$

Where $x_i \doteq (x_{i,1}, x_{i,2}, x_{i,3}, ..., x_{i,NG})^T \ (i = 1,2,3)$,

$x_{1,j} \doteq P_{o,j}$, $x_{2,j} \doteq S_{w,j}$, $x_{3,j} \doteq S_{g,j}$ $(j = 1,2,3,..., NG)$

$A \doteq (A_{i,j})_{3x3}$ is the coefficient matrix, $f \doteq (f_i)_{3x1}$ is the vector representing the right-hand side, and $x \doteq (x_i)_{3x1}$ is the vector of unknowns. And, x_1, x_2 and x_3 are oil phase pressure, water saturation, and gas saturation, respectively. Each $A_{i,j}$ is a heptadiagonal matrix ([4]).

This coupled linear system is significantly nonsymmetrical and highly indefinite it can be traced back to distinguished PDE behaviors of three types of unknowns. Fortunately, each $A_{i,j}$ is usually irreducible and diagonally dominant which let the whole matrix own some "good" property ([21]) and some ways can be found to decouple the whole system in order to solve the linear system more efficiently. We use Krylov subspace methods with combined preconditioning technique ([8]) to solve above linear system. In other words, Newton-Krylov-Schwarz algorithm is used in our solver, where Schwarz method is used to get our parallel version solver.

Several Krylov subspace methods were adopted to solve the linear systems. Algorithms proposed are GMRES, Othomin, BICGSTAB and GPBICG iterations.

GMRES ([14]) minimizes its residuals in a Krylov subspace at the price of more arithmetic comparing with other Krylov algorithms, but the implementation is robust. The convergence of GMRES is strictly monotonic on most cases. GMRES implementation is especially attractive for reservoir simulation applications. In fact, many current commercial reservoir simulators use this algorithm to solve the large-scale linear systems.

BICGSTAB ([15]) is a fast and smooth variant of the Bi-Conjugate Gradients method, while the GPBICG ([19]) is a generalized product-type method based on Bi-Conjugate Gradients method. Both the algorithms have been implemented in this project for solving non-symmetric linear systems. Unlike GMRES, which is based on the so-called Arnoldi orthogonalization process, both BICGSTAB and GPBICG are based on a bi-orthogonalization algorithm due to Lanczos, and therefore, they are intrinsically non-orthogonal.

Preconditioning is essential to efficiently solve a linear system. In the linear solver of the used simulator, the additive Schwarz preconditioning, iterative algorithm preconditioning, and ILU preconditioning are combined together in order to achieve adaptive and efficient parallel preconditioners for each problem.

Assume P is a projection operator, T_1 is an approximation of A^{-1}, and $P^T AP$ is invertible. It is possible to construct a preconditioner T_2 as $T_2 = P(P^T AP)^{-1} P^T$, and be straight forward to show the equality $P^T AT_2 = P^T$ The combined preconditioner following the idea in [8] may be defined as $B \doteq T_1 + T_2(I - AT_1)$ This preconditioner satisfies $I - AB = (I - AT_2)(I - AT_1)$, and it has the same form of multiplicative Schwarz algorithm ([22]). In fact, in order to decouple the coupled system and take full advantage of the "good" property of $A_{i,j}$, multiplicative Schwarz idea is used here. From another point of view, B also satisfies $P^T AB = P^T$. If B used as a right-preconditioner of Krylov subspace methods and the initial residual vector satisfies $P^T r^{(0)} = 0$, then $P^T r^{(k)} = 0$ $(\forall k \geq 1)$ will be retained during the course of iteration. In other words, the residuals $r^{(k)}$ are constrained by the used preconditioner B, and as a result, components of residual vectors can be assured to decrease to zero consistently.

The inverse matrix $(P^T AP)^{-1}$ may be computed by means of either a direct method or an iterative method, two types of combined preconditioners B exist, the one is based on ILU to solve $(P^T AP)^{-1}$, the other is based on GMRES instead, denoted by $PRE-ILU$ and $PRE-ITER$ respectively ([5]). Since parallel processing was used in the implementation, ILU was used only within local processors; additive Schwarz technique was used for inter-processor connection.

The preconditioned, parallel linear solvers use both domain decomposition and data parallelization techniques. At each Newton iteration step, the large-scale non-symmetric linear system is split across a number of processors. The split scheme is based on the size of grid, NX*NY*NZ. Currently, the number of gird cells in Z-direction needs to remain intact; the grid-partition is performed in both X- and Y-directions.

Each used Krylov subspace algorithm internally chooses a proper combined preconditioning, which has been optimally tuned for that algorithm. The combined preconditioning is also divided into several difficult levels, and the level can be dynamically changed during the simulation runs.

4. ALGORITHM EFFICIENCY AND COMPARISON

Some numerical experiments are used to demonstrate the relative efficiency of various Krylov subspace methods with two types of combined preconditioners.

Three check points exist in these experiments. First, the convergence behavior of Orthomin, GMRES, BICGSTAB, and GPBICG is examined with the same preconditioner. Second, the effect of two different preconditioners on the same Krylov subspace method is examined. Third, in order to obtain an effective linear solver for the simulator, different Krylov subspace methods with different preconditioners are applied to solve the same given problem ([4], [5]).

The experiments were performed on an 8 Pentium III 450 MHz PC cluster in RDCPS, each with 1GB memory, running under Slackware Linux. A 100Mb Ethernet switch was used for data communication.

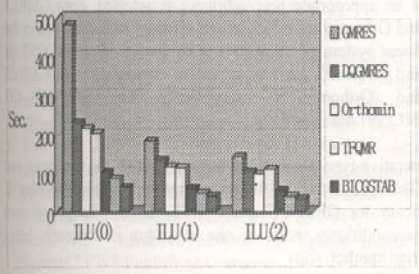
A typical PDE and three Black OiL Model practical problems are used to do the numerical tests. The equation of typical example as

$$-\frac{\partial}{\partial x}(e^{-xy}\frac{\partial u}{\partial x})-\frac{\partial}{\partial y}(e^{xy}\frac{\partial u}{\partial y})+2\alpha(\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y})$$

$$+\beta(2+\frac{1}{1+x+y})u=f$$

is solved by various Krylov subspace methods with ILU (k) preconditioner. Numerical results can be seen in Figure 1, which shows that ILU preconditioner is more suitable for BICG-type algorithm (such as CGS, BICGSTAB, GPBICG, BICGSTAB (l) etc.) than that of GMRES-type algorithm (such as GMRES, Orthomin, TFQMR, DQGMRES, FOM etc.).

Figure 1 : Comparison of Krylov Methods with ILU(k)
(a typical example, 256x256)



The first BLK model is a realistic data from an oil field in China, with a 35x44x34-grid system, 36 wells, the marching history is 4962 days. We test above three checkpoints by using the model and get experiences about preconditioned Krylov subspace methods. Numerical comparison of different Krylov methods with the same preconditioner *PRE-ILU* is given by Figure 2, different methods with *PRE-ITER* are given by Figure 3. The two Figures show that, BICGSTAB and GPBICG with PRE-ILU is obviously attractive comparing with GMRES. By choosing PRE-ITER, GMRES has the advantage of much shorter computational time compare with the other algorithms, while BICGSTAB and GPBICG lost their privilege.

The second BLK model is a dual porosity miscible flooding case, with a 130x123x10-grid system, 5 rock types, 10

Figure2: Comparison of Krylov Methods
with PRE-ILU (BLK Model, 35x44x34)

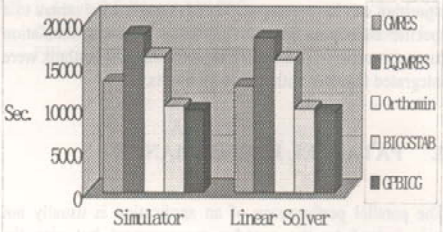
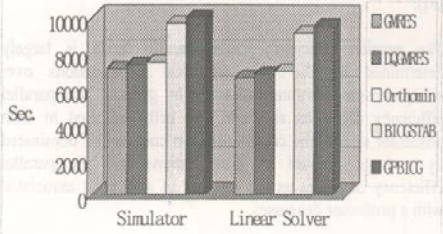
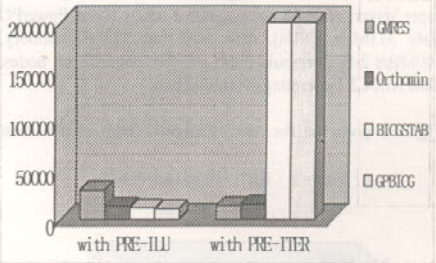


Figure3:Comparison of KrylovMethods
with PRE-ITER (BLK Model 35x44x34)



datum regions and 377 wells, the matching history is 31.5 years. This model is also used in the following parallel performance test. Figure 4 gives out its numerical test results. For this case, BICGSTAB and GPBICG with PRE-ITER behave much badly. Similar conclusions can also be drawn just as in the case of the first model.

Figure 4 : Parallel Linear Solver (4 CPU)
Elapsed Time (Sec.)



From Figures 2-4, some conclusions can be given. In the process of solving linear systems of black oil model, for Orthomin and GMRES, we think that the "optimal" preconditioner just looks like PRE-ITER, while for BICGSTAB and GPBICG, the "optimal" preconditioner maybe just like PRE-ILU. If comparing different Krylov methods with their "optimal" preconditioner, we may see that none of one algorithm is obviously better than another.

In fact, it is too difficult to choose an optimal preconditioner for specific iteration algorithm so as to adapt all cases of linear systems. Even for linear systems arisen from petroleum reservoir simulation, optimal preconditioners are also difficult to be constructed.

Meanwhile, these results show that all of the three algorithms proposed in this paper are suitable for oil reservoir simulation when appropriate preconditioners were selected. However, the convergence property of one algorithm can have computation advantages over others to a specific simulation problem Different reservoir simulation cases can achieve better performance when algorithms were integrated together with their own merits.

5. PATALLEL PERFORMANCE

The parallel performance of an application is usually not only judged by the speedup measurement but also the scaling measurement. Fixing the problem size and increasing the number of processors used measure speedup. Scaling is measured by fixing the local problem size in each processor and increasing the number of processors used ([4], [5], and [9]).

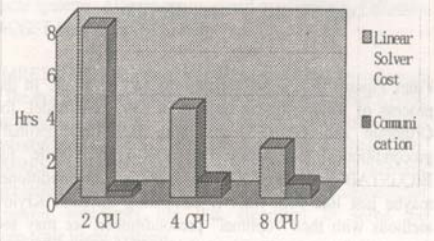
The parallel efficiency of a parallel code is largely determined by the ratio of local computations over interprocessor communications. In general, the parallel efficiency improves as more grid cells are used in each processor so that the communication cost can be dominated by a large amount of computations and the parallel efficiency degrades as the number of grid cells associated with a processor decreases.

In this part, two large field cases were selected to demonstrate the parallel performances.

The first case is the same as the second BLK model in 4th section. The performance test is on eight CPUs PC cluster in RDCPS. The second field case is a three-dimension and three-phase BLK data from some oil field in china, with $175 \times 176 \times 34$ grid system, 36 wells, 13 years of marching history, and the number of unknowns is 3141600. The used testing platform is Dawning 2000-II super server, with 80 computational nodes (Dual PowerPC 604e 333MHz CPUs), each node has 512MB memory, 100MB/s 2-D wormhole mesh network connects all nodes, IBM AIX4.2.1 operating system is used.

Figure 5 gives out the running elapsed times of the linear

Figure 5 : Multi CPU Running Results on PC Cluster



solver with 2,4,8 CPUs. Up to 8 CPUs, nearly linear speedup is obtained. From this figure, we can see that the communication costs increases when the number of CPUs increases. According to the analysis, the communication-to-computation ratio is roughly surface-to-volume apart from global reduction steps; however, well management of petroleum simulation inevitably increase a great number of global reductions due to production allocation and regulation plan. Consequently, if the well management project is complicated, lots of wells correlate to each other, the communication cost will increase dramatically. It means that many petroleum reservoir problems aren't scalable. The two cases we choose haven't complicated well management and good parallel performance obtained.

Figure 6 also shows excellent performance even when the number of CPUs increases to 64. The cost of communication amount here actually decreases. One reason is that additive Schwarz preconditioner lead to different total number of iterations when different number of CPUs is used. Another reason is related to the topology structure of the machine's network.

6. CONCLUSIONS

A parallel solution scheme has been developed to solve the sparse linear equations arising from finite difference discretization. The parallel implementation strategy is based on grid-partition and used on both shared and distributed memory parallel computer systems. The message passing protocol, MPI has been implemented into the solvers so that the solver is portable to systems that support this interface for interprocessor communications.

Several field cases were selected to show the effectiveness and efficiency of the linear solver. Results show that, up to eight CPUs, solver speedup ratio increases about linearly with the number of processors. The numerical experiments also show the solver is stable and robust to deal with complex geological structure and production scheme.

If an appropriate preconditioner is selected, both GMRES and Orhomin are effective and efficient methods to solve the linear system of oil reservoir simulation. Both BICGSTAB and GPBi-CG are also attractive comparing with GMRES and Orthomin.The convergence rate of GPBi-CG BICGSTAB is similar.

Iterative-type preconditioning is effective for Orthomin and GMRES, and ILU decomposition-type preconditioning is better for GPBi-CG and BICGSTAB.By using their own preconditioner, none of one algorithm is obviously better than another.

The parallel efficiency of a parallel code is largely determined by the ratio of local computations over interprocessor communications. The best parallel efficiency is achieved on the large numbers of grid cells, where the communication cost can be dominated by a large amount of computations.

REFERENCES

[1] D.W.Peaceman, "Fundamentals of Numerical Reservoir Simulation", Elsevier Scientific Publishing Company,1977

[2] K. Aziz, A.Settari, "Petroleum Reservoir Simulation", Applied Science Publishers LTD, London, 1979.

[3] Calvin C.Mattax, Robert L.Dalton, "Reservoir Simulation", H. L. Doherty Memorial Fund of AIME, SPE, Richardson, TX, 1990.
[4] Wei Liu, Jianwen Cao, Alberto Mezzatesta and Peng Zhu "Parallel Reservoir Simulation on Shared and Distributed Memory System", SPE 64797, 2000
[5] Jianwen Cao, Choi-Hong Lai, "Numerical Experiments of Some Krylov Subspace Methods for Black Oil Model", accepted for publication in Computers and Mathematics with Applications, CAM4793, Elsevier, 2001.
[6] Nolen, J.S., et al., "Reservoir Simulation on Vector Processing Computers," SPE Paper 9644 presented at the SPE Middle East Oil Technical Conference, Manama, Bahrain, March, 1981
[7] Wallis, J.R., et al., " A New Parallel Iterative Linear Solution Method for Large-scale Reservoir Simulation," SPE Paper 21209 presented at the 1991 SPE Symposium on Reservoir Simulation, Anaheim, California, Feb. 17-20, 1991
[8] Walis,J.R. "Incomplete Guassian Elimination as a Preconditioning for Generalized Conjugate Gradient Acceleration", SPE 12265, 1983.
[9] Abate, J., et al., "Parallel Compositional Reservoir Simulation on a Cluster of PCs, "Journal of Communications in Numerical Methods in Engineering, 1998
[10] Shiralkar, G.S., et al., "Falcon: A Production Quality Distributed Memory Reservoir Simulator," SPE Res. Eval. Eng., Oct. 1998
[11] Dogru, A.H., et al., "A Massively Parallel Reservoir Simulator for Large Scale Reservoir Simulation," SPE Paper 51886 presented at the 1999 SPE Symposium on Reservoir Simulation, Houston, Texas, Feb. 14-17, 1999
[12] Verdière, S., et al., "Applications of a Parallel Simulator to Industrial Test Cases," SPE Paper 51887 presented at the 1999 SPE Symposium on Reservoir Simulation, Houston, Texas, Feb. 14-17, 1999
[13] SimBest II User Guide, Baker Hughes Inc., 2000
[14] Saad, Y., and Schultz, M.H., "GMRES: a generalized minimal residual algorithm for solving nonsymmetrical linear systems," SIAM Journal on Scientific and Statistical Computing. 7:856-869,1986.
[15] Van der Vorst, H.A., "Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," SIAM Journal on Scientific and Statistical Computing.12:631-644,1992.
[16] Saad,. Y., " Iterative Methods for Sparse Linear Systems," PWS Publishing Company, 1995.
[17] Nachtigal, N. M. et al., "How fast are nonsymmetric matrix iterations?" SIAM J. Matrix Anal.Appl. 13:778-795, 1992.
[18] Sun, Jiachang, and Cao, Jianwen, " A Class of Local Green-like Parallel Preconditioner Algorithm for Elliptic Discrete Equations II: for Non Self-conjugate Equations," Mathematica Numerica Sinica, Vol.18, No.2, 1995.
[19] Zhang, Shao-Liang " GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems," SIAM Journal on Scientific and Statistical Computing. Vol.18 No.2 PP537-551, March 1997.
[20] Cao, Jian-Wen, "Numerical Experiments and Analyses of Krylov Subspace Methods on One Kind of PDE," Journal On Numerical Methods and Computer Applications. Vol.20 No.4 1999.
[21] H.Klie,M.Rame & M.Wheeler, "Two-stage Preconditions for Inexact Newton Methods in Multi-phase Reservoir Simulation", CRPC-TR96660, Rice University,1996.
[22] B.Smith,P.E.Bjorstad,W.D.Gropp, "Domain Decomposition – Parallel Multilevel Methods for Elliptic Partial Differential Equations", 1996.

Sequential Approximation to Virtual Boundary for Parallelization of Hybrid-SRM Scheme*

Chen Xianqiao Guo Qingping
College of Computer Science & Technology, Wuhan University Technology, Wuhan 430063

Lin Ping
Department of Mathematics, The National University of Singapore, Singapore 117543

ABSTRACT

Hybrid Scheme is a important formulation in solving Navier-Stokes equations (incompressible flow problems), and SRM algorithm keeps the benefits of the penalty method, that is, velocity and pressure can be obtained separately and no pressure-Poisson equation is involved, unlike the penalty method the SRM is more stable, less stiff. In then case that a large number of time steps are needed, we introduce Domain-Decomposition based parallel techniques, and apply Sequential Approximation Virtual Boundary to compute the internal boundaries of sub-domains, less iterative is needed., computational results shows that the result is very well, and the speedup ratio of our method is larger .

Keywords: Hybrid Scheme, Sequential Regularization Method (SRM), Sequential Approximation Virtual Boundary (SAVB), Parallel algorithm, Domain-Decomposition

1. INTRODUCTION

The past twenty of thirty years have witnessed a great deal of progress in the area of computational fluid dynamics. A large number of methods has been proposed for the numerical solution of the problem. Direct discretization includes finite difference and finite volume techniques, mixed finite element methods using conformal and non-conformal elements and spectral methods. To increase the stability and to reduce the computational cost reformulation and /or regularization of the equations are normally needed. Examples of such methods include pseudo-compressibility methods, penalty methods, projection and pressure-Poisson formulations. Among them, the penalty method is important, since its formulation is very simple and artificial boundary values for the pressure are not required. However, an obvious drawback is that it results in a very stiff problem in the time variable, so that explicit time discretization is not possible to be used.

Another topic of great recent interest is the numerical solution of differential-algebraic equations. In their most popular special forms, these are ordinary differential equations with some equality constraints. Recall that an important concept for measuring the difficulty in solving DAE is given by the (differential) index, which is defined by the minimal number of analytical constraint differentiation such that the DAE can be transformed by algebraic manipulations into an explicit first-order differential system for all original unknowns.

While a significant body of knowledge about the theory and numerical methods for DAE has been accumulated, not much has been extended to partial differential-algebraic equations (PDAE). The incompressible Navier-Stokes equations form, in fact, an example of a PDAE, these equations read

$$u_t + (u \cdot grad)u = \mu \Delta u - gradp + f \qquad (1.a)$$

$$divu = 0 \qquad (1.b)$$

$$u|_{\partial\Omega} = b, u|_{t=0} = a \qquad (1.c)$$

The system (1) can be seen as a partial differential equation with constraint (1.b) with respect to the time variable t. Indeed, the pressure-Poisson reformulation of (1) corresponds to a direct index reduction for the PDAE, i.e. a differentiation of the constraint with respect to t followed by substitution into the momentum equations. Relationship between reformulation or sequential regularization of DAE and Navier-Stokes equations has been discussed. Then a sequential regularization method (SRM), which is a generalization from its DAE version, was proposed and analyzed for the Navier-Stokes equations in that paper. We are interested in the SRM because it keeps the benefits of the penalty method but, unlike the penalty method, the regularization parameter can be chosen relatively large and the accuracy of the regularization formulation can be achieved by a few iterations. Therefore, the regularized problems are more stable or less stiff. Hence, more convenient (non-stiff) methods can be used for time integration, i.e. explicit time discretization satisfying usual time step restrictions can be used. This property is especially attractive when we solve nonlinear problems such as Navier-Stokes equations since there is no need to solve nonlinear algebraic systems. We will see later that it is more efficient for parallel computation due to much less internal-processor communication load for both velocity and pressure components. So the implementation and computation can be largely simplified and reduced. Moreover

* The work was partially supported by the National Science Foundation of China (NSFC) under grant No. 69773021
The work of the first author was partially supported under Singapore academic research grant R-151-000-013-112. The work of the second author was partially supported under Singapore academic research grant R-146-000-016-112
The work of the third author is supported under Singapore academic research grants R-146-000-016-112 and R-151-000-013-112.

the time step restriction associated with the explicit time discretization may be loosened in the case of high Reynolds number for the Navier-Stokes equations.

The present study represents an effort to apply the new regularization method to a real model flow problem. Although more and more sophisticated techniques are developed we demonstrate that very simple discretization works very well under the new regularization formulation for the problem. Moreover, a Domain-Decomposition-based parallelization with Sequential Approximation Virtual Boundary for Parallelization Hybrid-SRM Scheme is studied and successfully incorporated into the new regularization method

2. THE SEQUENTIAL REGULARIZATION METHOD (SRM) FOR THE NAVIER-STOKES EQUATIONS

It is well known that for DAE, if the constraints are not treated carefully to reduce the index then they need not be satisfied when then problem is integrated in time. Baumgarte's stabilization is a popular method which, when applied to the Navier-Stokes equations (1), corresponds to replacing the incompressibility constraint by the following

$$\alpha_1(divu)_t + \alpha_2 divu = 0 \tag{2}$$

The importance of the treatment of the incompressibility constraint was also recognized in the Navier-Stokes context. $\alpha_1 = 1$ and $\alpha_2 = 0$ corresponds to the well known pressure-Poisson reformulation. If we apply a penalty idea to the new constraint (2) we then obtain

$$-\varepsilon(p - p_0) = \alpha_1(divu)_t + \alpha_2 divu, \tag{3}$$

Where ε is a small constant and p_0 is an initial guess of the pressure. Solving the coupled system (1), (2) and replacing p_0 by p the newly obtained p recursively, we obtain the sequential regularization method:

With $p_0(x,t)$ an initial guess and $\alpha_1, \alpha_2 \geq 0$, for s=1,2... solve the problem

$$\varepsilon(u_s)_t - grad(\alpha_1(divu_s)_t + \alpha_2 divu_s) + \varepsilon(u_s \cdot grad)u_s = \varepsilon\mu\Delta u_s - \varepsilon grad p_{s-1} + \varepsilon f, \tag{4.a}$$

$$u_s|_{\partial\Omega} = b, u_s|_{t=0} = a, \tag{4.b}$$

$$p_s = p_{s-1} - \frac{1}{\varepsilon}(\alpha_1(divu_s)_t + \alpha_2 divu_s) \tag{4.c}$$

It is proved in that if $\alpha_1 \neq 0$ then

$$u - u_s = O(\varepsilon^s) \tag{5}$$

in the sense of $\|\cdot\|_{H_1}$ and

$$p - p_s = O(\varepsilon^s) \tag{6}$$

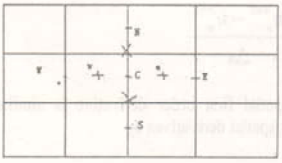
in the sense of $\int \|\cdot\|_{L_2}^2 dt)^{\frac{1}{2}}$, where s=1,2,---.

In the rest of the paper we will focus on a model fluid flow problem –driven cavity flow. In the spatial direction we will use Hybrid difference schemes when the Reynolds number is high. In the temporal direction we will use implicit difference schemes for $\alpha_1 \neq 0$. It is always troublesome to use iterative methods to solve the resulting nonlinear system from the discretization of the Navier-Stokes equations. We would like to use more reliable approximate line-box system solvers. The storage limit of computers would not allow having a relatively fine grid. We thus apply domain decomposition idea. Meanwhile the domain-decomposition-based parallelization would reduce the computational time.

The SRM relates to the idea of penalty methods but, unlike the penalty method, the regularized problems are less stiff as we mentioned above. Hence, more convenient methods can be used for time integration, and then nonlinear terms can be treated easily. The SRM is based on the penalty method. Hence it does not require artificial boundary conditions for the pressure p. So it is more natural than various pressure-Poisson formulations. Moreover, **u** and p are calculated separately. Actually p can be easily obtained by substitution from (3).

3. APPLICATION TO NAVIER-STOKES EQUATIONS FOR SHEAR-DRIVEN CAVITY FLOW

Then incompressible flow in a square cavity whose bottom wall moves with a uniform velocity in its own plane has



served over and over again as a model problem for testing and evaluating numerical techniques, in spite of the singularities at two of its corners. The governing equations of the cavity-flow problem are

$$u_t + uu_x + vu_y = -p_x + \frac{1}{\mathrm{Re}}(u_{xx} + u_{yy}), \tag{7.a}$$

$$v_t + uv_x + vv_y = -p_y + \frac{1}{\mathrm{Re}}(v_{xx} + v_{yy}), \tag{7.b}$$

$$u_x + v_y = 0 \tag{7.c}$$

Here, Re is the Reynolds number. The boundary conditions are

$(u,v)^T = (0,0)^T$ on top, two sides of cavity

$(u,v)^T = (1,0)^T$ on the bottom of cavity

The start of the flow is impulsive: $(u,v)^T = (0,0)^T$, for t=0, then the value of (u, v) is jumped instantaneously up to (1, 0) on the bottom to start the flow. The SRM formulation for the

cavity-flow equations reads

$$\frac{\partial}{\partial t}(\varepsilon u^s - \alpha_1(u^s_{xx} + v^s_{yx})) =$$

$$\varepsilon(\frac{1}{\mathrm{Re}}\Delta u^s - p^{s-1}_x - (u^s u^s_x + v^s u^s_y)) + \alpha_2(u^s_{xx} + v^s_{yx}) \quad (8.a)$$

$$\frac{\partial}{\partial t}(\varepsilon v^s - \alpha_1(u^s_{xy} + v^s_{yy})) =$$

$$\varepsilon(\frac{1}{\mathrm{Re}}\Delta v^s - p^{s-1}_y - (u^s v^s_x + v^s v^s_y)) + \alpha_2(u^s_{xy} + v^s_{yy}) \quad (8.b)$$

$$p^s = p^{s-1} - \frac{1}{\varepsilon}(\alpha_1\frac{\partial}{\partial t} + \alpha_2)(u^s_x + v^s_y) \quad (8.c)$$

4. HYBRID DIFFERENCE SCHEME FOR THE REGULARIZED PROBLEM

Let $\Delta t, \Delta x, \Delta y$ denote step sizes in time and spatial directions, respectively. Thus, grid points can be expressed as

$x_i = i\Delta x, i = 0,1,\cdots, N_x; y_j = j\Delta y, j = 0,1,\cdots, N_y;$

$t_n = n\Delta t, n = 0,1,\cdots, N_t.$

Replacing spatial first order derivatives by

$$\frac{\partial u}{\partial x} \approx \frac{u^{n+1}_e - u^{n+1}_w}{\Delta x} \quad (9)$$

The other spatial first order derivative is similar to (9). Second order spatial derivatives by

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \approx \frac{(\frac{\partial u}{\partial x})^{n+1}_e - (\frac{\partial u}{\partial x})^{n+1}_w}{\Delta x} + \frac{(\frac{\partial u}{\partial y})^{n+1}_n - (\frac{\partial u}{\partial y})^{n+1}_s}{\Delta y}, \quad (10)$$

For the temporal discretization, we use Euler scheme, we thus have

$$\frac{\partial u}{\partial t} = \frac{u^{n+1}_c - u^n_c}{\Delta t} \quad (11)$$

Substituting into (8), we obtain the following formulations:

$$A_E u_E + A_W u_W + A_N u_N + A_S u_S - A_C u_C + D_W P_W - D_C P_C = S_u$$

$$B_E v_E + B_W v_W + B_n v_n + B_S v_S - B_C v_C + D_N P_N - D_C P_C = S_v$$

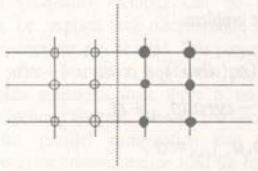
5. PARALLELIZATION VIA DOMAIN DECOMPOSITIONS

We have observed that if $\alpha_1 \neq 0$ we need to solve a linear system of size $(2N_xN_y)\times(2N_xN_y)$. The linear system is associated to the operator $I + \frac{\alpha_1}{\varepsilon} graddiv$. There are discussions on preconditioning techniques for this operator. Frequently iterative methods do not work well for algebraic systems resulting from the Navier-Stokes equations. Meanwhile, because of the large size of the system, a direct solver is generally impossible due to the large cost of the storage and computation. We thus consider incorporating parallel techniques into our method. Domain-Decomposition based parallel techniques can be used to effectively reduce the size linear system and the time of computation.

If we have k processors, we can divide the domain into k sub-domains. Hence when we use one processor for one sub-domain, the size of the linear system it solves is reduced by a factor of $\frac{1}{k^2}$. Hence, direct linear system solvers are possible to be used even if the grid for the original domain is relatively fine. Nevertheless, since we do not know the boundary values at the internal virtual boundaries of sub-domains, some iterative procedure is still needed.

Let us consider two neighboring domains in a partition. In Figure, variables defined on white points are calculated on one processor, and those on black points are on another. Algebraic equations corresponding to the points at the internal boundary (virtual boundary) require variable values belong to the neighboring domain which may be obtained by message passing. Since we don't know variable values at the internal boundary we have to repeat the message passing a few times until the difference of two consecutive iterative values is reduced to be less than a tolerance.

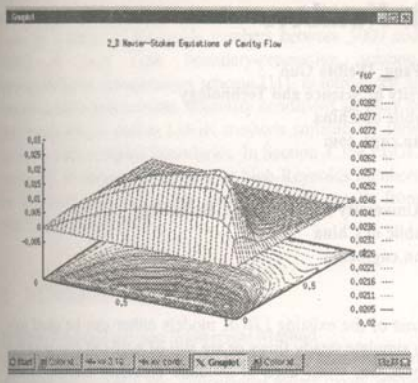
If we only exchange the information near the boundary, that will form a chimb between the two sub-domains, and convergence is very slow, then some times the parallel algorithm cost more times than serial algorithm. So we make sequential points $\{x_1, x_2, \cdots, x_n\}$ in each sub-domains, $x_n \to x_b, x_b$ is the internal boundary (virtual boundary), at first time iteration, we use x_1, and then x_2,---. Here x_1 is the center point in the sub-domains.

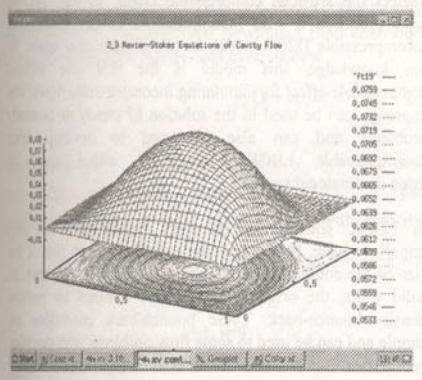


6. COMPUTATION RESULTS

In this section we have done some computations for 2D square cavity flow with a cluster of PC's running PVM as a message passing tool. We give two figures to show the computation results respectively represent using Sequential Approximation Virtual Boundary and no using Sequential Approximation

Virtual Boundary.





REFERENCES

[1] Guo Qingping, Y. Paker, et al, Optimum Tactics of Parallel Multi-grid Algorithm with Virtual Boundary Forecast Method Running on a Local Network with the PVM Platform, Journal of Computer Science and Technology, July 2000, Vol.15, No.4, pp355~359

[2]. L.R. Matheson et al, Parallelism in Multigrid Methods: How Much is too Much, International Journal of Parallel Programming, Vol.24, No.5, Plenum Publishing Corporation, 1996.

[3].Wang xianfu & Zhang Shesheng, Hydrofoil supercave flow with free surface. Jour. of Shipbuilding. No.4 (1996). pp1-8.

[4] Fan Yingquan. Advance mathematics. 1962, Beijing. pp276~312.

[5]. Wei Jianing, Guo Qingping et al., A Multigrid Parallel Algorithm of One Dimensional Virtual Boundary Ransack Forecast, Journal of Wuhan Transportation University, Wuhan China, Vol. 24 No. 2, April 2000, pp108~112

LBGK Simulation of Driven Cavity Flow At High Reynolds Numbers

Baochang Shi, Nengchao Wang, Weibin Guo
Parallel Computing Institute, Huazhong University of Science and Technology
Wuhan 430074, People's Republic of China
Email: sbchust@public.wuhan.cngb.com
and
Zhaoli Guo
National Laboratory of Coal Combustion, Huazhong University of Science and Technology
Wuhan 430074, People's Republic of China
Email: pcihust@public.wuhan.cngb.com

ABSTRACT

Results for viscous, incompressible two-dimensional driven cavity flows were investigated by using a newly developed incompressible LBGK model with a robust boundary-processing scheme with second-order accuracy. A detailed analysis for a range of Reynolds numbers between 5000 and 20000 is presented. Thorough comparisons with other solutions show that the model gives accurate results over a wide range of Reynolds numbers.

Keywords: Driven Cavity Flow, Incompressible LBGK Model, Nonequilibrium Extrapolation.

1. INTRODUCTION

Today, despite enormous progress in CFD, limitations still exist because of computer resources. It is apparent that several orders of magnitude improvement in both speed and memory are necessary to solve problems of contemporary interest. These requirements are obtained assuming today's solution algorithms and computer architecture. Since the technologies of scalar and vector computing have had substantial development, further work is unlikely to yield significant increases in computer performance. Massive parallel processing, on the other hand, appears to possess the capability to partially fill the gap between computational needs and present supercomputer performance. The efficient use of massively parallel computers requires new parallel algorithms. The lattice Boltzmann BGK (LBGK) method is such a fully parallel algorithm [1].

The LBGK method is a relatively new approach that uses simple microscopic models to simulate macroscopic behavior of transport phenomena. Compared with conventional CFD approach, its algorithm is simple, fast, and intrinsically parallel. It is also easy to incorporate complicated boundary conditions. So the LBGK method is more computationally efficient using current parallel computer [1,2]. The LBGK models commonly used in the solution of the incompressible Navier-Stokes equations can be viewed as compressible schemes to simulate incompressible fluid flows, and there is the *compressible effect*, which might lead to some undesirable errors in numerical simulations. Some LBGK models have been proposed to reduce or eliminate such errors [3-6]. However, most of the existing LBGK models either can be used only to simulate steady flows or are still of artificial compressible form. So, when used to simulate unsteady incompressible flows, these methods require some additional conditions to neglect the artificial compressible effect. In Ref. [7], we have proposed a 9-bit incompressible LBGK model, the incompressible D2G9 model, in two-dimensional space. To our knowledge, this model is the first one without compressible effect for simulating incompressible flows. The approach can be used in the solution of steady or unsteady problems and can also be used to develop other incompressible LBGK models in either two- or three-dimensional space.

In LBGK simulations, boundary condition is a very important issue. Proper boundary conditions can reduce the size of the computational domain and decrease the cost. At solid walls, the original schemes are realized by particle density bounce-back. These bounce-back conditions are simple and can be used to some flow problems with complex geometries. But it is known that bounce-back wall boundary conditions are of first-order accuracy and cannot process some complex boundary conditions, such as kinematics boundary condition and Neumann boundary condition. To avoid of these problems, several new type boundary-processing schemes have been proposed and improved the overall accuracy of LBGK methods [8]. But, these schemes are imposed certain restrictions. Although the extrapolation scheme proposed by Chen *et al.* used second-order extrapolation, which is consistent with LBGK methods, we found that the second-order extrapolation scheme has poor stability for high Reynolds numbers. It is necessary to establish a boundary-processing scheme, which is of higher order accuracy, has robust stability and is efficient for arbitrary complex geometric boundaries.

In various flow problems, the driven cavity flow has served as prototypes of complex flow problems for testing and evaluating numerical schemes due to its rich vortex structures and simple geometry. Numerical simulations for the driven cavity flow have been studied by many authors using traditional schemes such as finite difference, multigrid method, and finite element methods and their variation [9,10]. Hou and Zou performed the lattice Boltzmann simulation with detailed analysis [2]. However, most work was done at low or moderate Reynolds numbers, thus there is a significant lack of information for high Reynolds numbers. When Reynolds number exceeds 5000, the main difficulties encountered of these methods are increasing CPU times and the cost of computation because of large size of

* This work is supported by the National Science Foundation of China (Grant 60073044).

the computational grid employed, instability or unserviceable of schemes. For validation of our model, as an application of the D2G9 LBGK model in classical fluid mechanics, we investigate two-dimensional driven cavity flows for a range of Reynolds numbers between 5000 and 20000. A new type boundary-processing scheme, non-equilibrium extrapolation scheme [11], is introduced in Section 2. With this scheme, boundary conditions are easy to implement, which makes LBGK methods suitable for flow problems with complex boundaries. In Section 3, the LBGK simulation of driven cavity flow at high Reynolds numbers are discussed and thoroughly compared with results from other drawn in Section 4.

2. NUMERICAL DETAILS

Problem Description and Numerical Method

The configuration of the driven cavity flow considered in this paper consists of a two-dimensional square cavity whose top plate moves from left to right with a uniform velocity, while the other three walls are fixed (see Fig.1). The flow is assumed to be two-dimensional Newtonian fluid described by the incompressible Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \upsilon \nabla^2 \mathbf{u} \tag{2}$$

where $\mathbf{u} = (u, v)$ is the velocity vector, $p = c_s^2 \rho$ is the pressure ($c_s = c/\sqrt{3}$, the sound speed), and υ is the kinematics viscosity determined by

$$\upsilon = \frac{(2\tau - 1)}{6} \frac{(\Delta x)^2}{\Delta t} \tag{3}$$

In Eq. (3), Δx , Δt , and τ are the lattice grid spacing, the time step and the dimensionless relaxation time, respectively.

The incompressible D2G9 LBGK equation [7] of the system is defined by equation (4):

$$g_i(\mathbf{x} + c\mathbf{e}_i \Delta t, t + \Delta t) - g_i(\mathbf{x}, t) = -\frac{1}{\tau}[g_i(\mathbf{x}, t) - g_i^{(0)}(\mathbf{x}, t)] \tag{4}$$

where subscript i indicates the velocity direction, and $c = \Delta x / \Delta t$ is particle velocity. $g_i(\mathbf{x}, t)$ is the distribution function at node $\mathbf{x}$ and time t with velocity $\mathbf{e}_i$, and $g_i^{(0)}(\mathbf{x}, t)$ is the corresponding equilibrium distribution defined by equation (5):

$$g_0^{(0)} = -4\sigma \frac{p}{c^2} + s_0(\mathrm{u}) \ , \ g_i^{(0)} = \lambda \frac{p}{c^2} + s_i(\mathrm{u}) \ \ i = 1,2,3,4 \ , \ g_i^{(0)} = \gamma \frac{p}{c^2} + s_i(\mathrm{u}) \ \ i = 5,6,7,8 \tag{5}$$

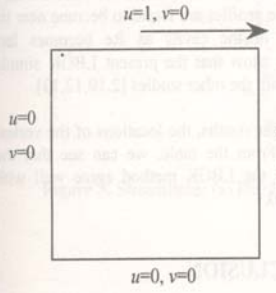


Figure 1. Configuration of flow problem

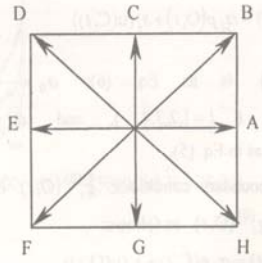


Figure 2. Nodes on boundary

Where σ, λ , and γ are parameters satisfying $\lambda + \gamma = \sigma, \lambda + 2\gamma = 1/2$, $s_i(\mathrm{u})$ is

$$s_i(\mathrm{u}) = \omega_i [3\frac{(\mathbf{e}_i \cdot \mathrm{u})}{c} + 4.5\frac{(\mathbf{e}_i \cdot \mathrm{u})^2}{c^2} - 1.5\frac{|\mathrm{u}|^2}{c^2}] \tag{6}$$

with weights $\omega_0 = 4/9$, $\omega_i = 1/9$ $(i = 1,2,3,4)$, and $\omega_i = 1/36$ $(i = 5,6,7,8)$.

The velocity and pressure of flow are given by

$$\mathrm{u} = \sum_{i=1}^{8} c\mathbf{e}_i g_i \ \text{ and } \ p = \frac{c^2}{4\sigma}[\sum_{i=1}^{8} g_i + s_0(\mathrm{u})] .$$

Having chosen the appropriate lattice size and the characteristic velocity, the viscosity, υ, can be calculated for a given Reynolds number and then the time step can be determined by using Eq.(3). Starting from initial velocity and pressure fields, the distribution function g_i, velocity u, and pressure p, etc. can be computed using above equations. The iterative procedure can be terminated when certain criterion is reached, and then the resulting flow field is obtained.

Boundary Conditions

The basic idea of the nonequilibrium extrapolation scheme [11] is simple. We split the distribution function into two parts at per node of boundary: equilibrium and nonequilibrium. A new type equilibrium distribution function is introduced to approximate the equilibrium

distribution in term of the specific boundary conditions, while nonequilibrium distribution is determined by nonequilibrium extrapolation. Thus the distribution function defined is of second-order overall accuracy and the boundary schemes designed have strong stability [11].

For each time step in D2G9 LBGK, the updating of the particle distribution can be split into two sub steps: collision and streaming. They are defined by

$$g_i^+(\mathbf{x},t) = (1-\frac{1}{\tau})g_i(\mathbf{x},t) + \frac{1}{\tau}g_i^{(0)}(\mathbf{x},t) \quad (7)$$

and

$$g_i(\mathbf{x}+c\mathbf{e}_i\Delta t, t+\Delta t) = g_i^+(\mathbf{x},t) \quad (8)$$

Where the equilibrium distribution function $g_i^{(0)}(x,t)$ is the same form as Eq. (5).

In Figure 2 assuming that EOA reside on the boundary and DCB, FGH are the inner and outer of flow field, respectively. After streaming process, let the system time be t. To obtain the distribution function at node O, we decompose the function $g_i(\mathrm{O},t)$ into two parts:

$$g_i(\mathrm{O},t) = g_i^{(0)}(\mathrm{O},t) + \varepsilon g_i^{(1)}(\mathrm{O},t) \quad (9)$$

where $\varepsilon g_i^{(1)}(\mathrm{O},t)$ is nonequilibrium distribution function at node O, and $\varepsilon \sim \Delta x \sim \Delta t$.

When computing $g_i^{(0)}(\mathrm{O},t)$, for pressure boundary conditions, we use correct equilibrium distribution function $\bar{g}_i^{(0)}(\mathrm{O},t)$ as follows to approximate $g_i^{(0)}(\mathrm{O},t)$:

$$\bar{g}_i^{(0)}(\mathrm{O},t) = \alpha_i p(\mathrm{O},t) + s_i(\mathrm{u(C},t)) \quad (10)$$

where $s_i(\mathrm{u})$ is as Eq. (6); $\alpha_0 = -4\sigma/c^2$, $\alpha_i = -\lambda/c^2$ ($i = 1,2,3,4$), and $\alpha_i = -\gamma/c^2$ ($i = 5,6,7,8$) as in Eq. (5).

For velocity boundary conditions, $\bar{\bar{g}}_i^{(0)}(\mathrm{O},t)$ is used to approximate $g_i^{(0)}(\mathrm{O},t)$ as follows:

$$\bar{\bar{g}}_i^{(0)}(\mathrm{O},t) = \alpha_i p(\mathrm{C},t) + s_i(\mathrm{u(O},t)) \quad (11)$$

The non-equilibrium distribution function with second-order accuracy, $\varepsilon g_i^{(1)}(\mathrm{O},t)$, is determined by

$$\varepsilon g_i^{(1)}(\mathrm{O},t) = g_i(\mathrm{C},t) - g_i^{(0)}(\mathrm{C},t) \quad (12)$$

3. NUMERICAL RESULTS

Numerical simulations are carried out using the methods presented above for the driven cavity flow with Re=5000, 7500, 10000, 15000, and 20000, respectively, on 256×256 lattice. The Reynolds number used in the problems is $\mathrm{Re} = LU/\upsilon$, where L is the height or width of the cavity, U is the uniform velocity of the top plate. The relaxation parameter $w = 1/\tau$ is set to be 1.85, 1.92, 1.95, 1.95, 1.96, respectively. (σ,λ,γ) in Eq.(5) is set to be (5/12,1/3,1/12) such that $(\lambda,\gamma) = 3\times(\omega_1,\omega_5)$ which has the symmetry to agree with the method, and we find that the simulations with this set are more robust. For the walls, no-slip boundary conditions were prescribed by the nonequilibrium extrapolation method given above. The flow with Re=5000 is first simulated, where the initial condition is set as p=0, and the velocities at all nodes, except the top nodes, are set as $u = v = 0$. The simulations for Re=7500, 10000, 15000, and 20000 start from the steady states for Re=5000, 7500, 10000, and 15000, respectively. In the simulations, steady states for $\mathrm{Re} \le 10000$, and $\mathrm{Re} > 10000$ are reached when the difference between the velocities at the center of the cavity for successive 1000 steps are less than 5×10^{-6}, and 5×10^{-4}, respectively.

Figure 3 shows the contours of the stream function of the flows for the Reynolds numbers considered. These plots show clearly the effect of the Reynolds number on the flow pattern. At Re=5000, in addition to the primary, center vortex, and three first-class vortices, a pair of secondary ones of much smaller strength develop in the lower corners of the cavity. When Reynolds number reaches to 7500, a tertiary vortex appears in the lower right corner. Stationary solutions were found for Reynolds numbers up to 10000. We can also see that the center of the primary vortex moves toward the geometric center of the cavity as the Reynolds number increases and becomes fixed in x-direction. As the Reynolds number increases, no more steady solution was found and the flow becomes periodic in time (a period is about 2000 time steps, but further study is needed). Here only one mode is given for Re=15000 and 20000, respectively. The velocity components, u and v, along the vertical and horizontal centerline for different Re are shown in Fig. 4. The profiles are found to become near linear in the center core of the cavity as Re becomes large. These observations show that the present LBGK simulation is in agreement with the other studies [2,10,12,13].

To quantify the results, the locations of the vortex are listed in Table I. From the table, we can see that these values predicted by the LBGK method agree well with those of previous work.

4. CONCLUSION

With detailed studies of the cavity flow problem, we were able to show that our implementation of LBGK method yields reliable results using the same mesh size. By using the incompressible D2G9 LBGK model, the compressible effect is eliminated efficiently. The proper implementation of the boundary conditions is crucial for the LBGK simulation. Nonequilibrium extrapolation method has robust stability and the overall accuracy of distribution function is of second order.

In LBGK simulation, since the maximum velocity and lattice size are limited, the relaxation parameter, w, needs to be large to achieve the higher Reynolds numbers. It is found that the lowest c leading to stable simulations depends on the ratio of the particle velocity, c, and the physical velocity. The appropriate ratio is about 3~10. On the other hand, to obtain a reliable simulation, w should not be too close to its upper limit. It cannot be larger than 2 to ensure positive viscosity.

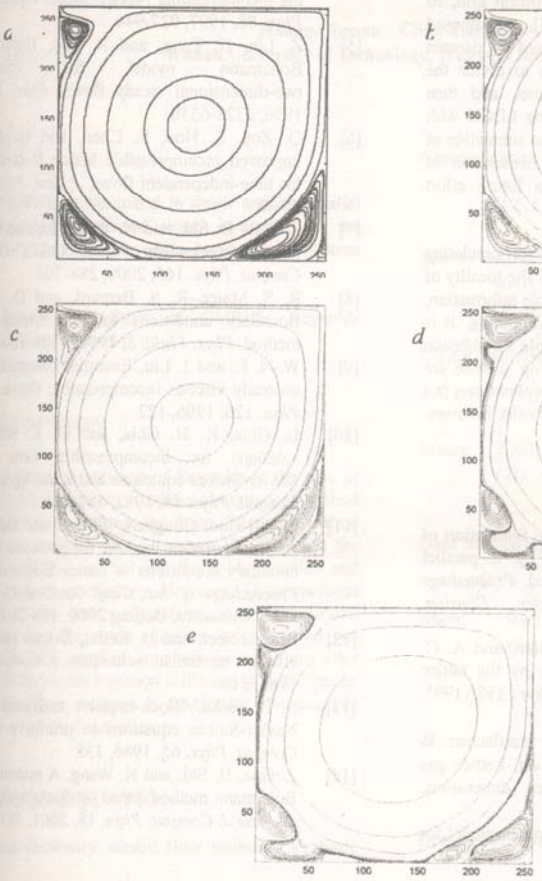


Figure 3. Streamlines: (a) Re=5000; (b) Re=7500; (c) Re=10000; (d) Re=15000; (e) Re=20000

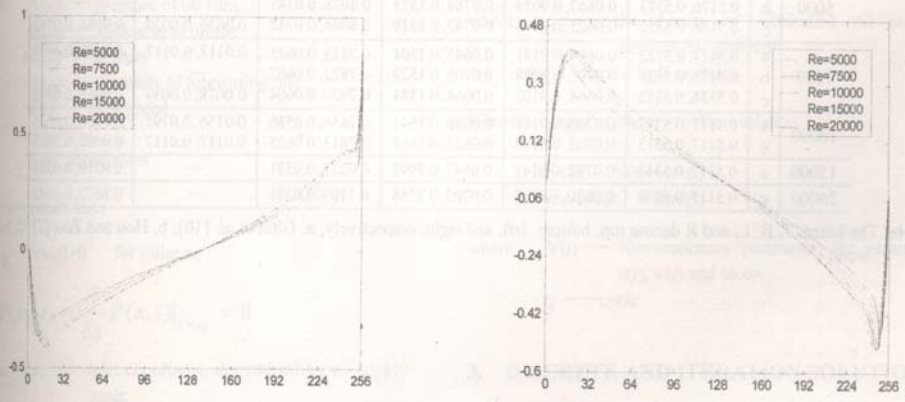


Figure 4. The velocity components, u (left) and v (right), along the vertical and horizontal lines through the cavity centers for the driven cavity flow for different Reynolds numbers.

The LBGK simulation in this paper uses uniform grid, so some local details can't be found. In Ref. [14], we proposed a LBGK model using nonuniform mesh based on domain decomposition technique. The basic idea is to divide the flow domain into some regular subdomains, and then perform simulations on each subdomain using LBGK with uniform grid. We will further use this model to simulation of the driven cavity flows. Furthermore, the simulation of three-dimensional driven cavity flows is a future effort direction.

LBGK method is a relatively new approach for simulating complex flows. It is parallel in nature due to the locality of particle interaction and the transport of particle information, so it is well suited to massively parallel computing. It is apparent that further design on both suitable equilibrium distribution function and boundary processing scheme are needed. Applying LBGK method to other complex flows is a challenging work, which is our main research work in future.

REFERENCES

[1] S. Hou, Q. Zou, and A. C. Cogley, Simulation of three-dimensional turbulent flow by a parallel algorithm, lattice Boltzmann method, *Proceedings of the ASME Heat Transfer Division*. HTD-Vol.317-2, 1995, 451-467

[2] S. Hou, Q. Zou, S. Chen, G. Doolen, and A. C. Cogley, Simulation of cavity flow by the lattice Boltzmann method, *J. Comput. Phys*. 118, 1995, 329-347

[3] U. Frish. D. d'Humi é res, B. Hasslacher, P. Lallemand, Y. Pomeau, and J.-P.Rivet, Lattice gas hydrodynamics in two and three dimensions, *Complex Syst.*1, 1987, 649-707

[4] X. He and L.-S. Luo, Lattice Boltzmann model for the incompressible Navier-Stokes equation, *J. Stat. Phys*. 88, 1997, 927-944

[5] Z. Lin, H. Fang, and R. Tao, Improved lattice Boltzmann model for incompressible two-dimensional steady flows, *Phys. Rev. E*. 54, 1996, 6323-6330

[6] Q. Zou, S. Hou, S. Chen, and G. Doolen, An improved incompressible lattice Boltzmann model for time-independent flows, *J. Stat. Phys*. 81, 1995, 35-48

[7] Z. Guo, B. Shi, and N. Wang, Lattice BGK model for incompressible Navier-Stokes equation, *J. Comput. Phys*. 165, 2000, 288-306

[8] R. S. Maier, R. S. Bemard, and D. W. Grunau, Boundary conditions for the lattice Boltzmann method, *Phys. Fluid*. 8, 1996, 1788-1801

[9] W.-N. E, and J. Liu, Essential compact scheme for unsteady viscous incompressible flows, *J. Comput. Phys*. 126, 1996, 122

[10] U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys*. 48, 1982, 387-411

[11] Zhaoli Guo, Chuguang Zheng, and Baochang Shi, An extrapolation method for pressure and velocity boundary conditions in lattice Boltzmann method, *Proceedings of Int. Conf. Applied Computational Fluid Dynamics*, Beijing 2000, 198-202

[12] R. Schreiber, and H. Keller, Driven cavity flow by efficient numerical techniques, *J. Comput. Phys*. 49, 1983, 310

[13] S. P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, *J. Comput. Phys*. 65, 1986, 138

[14] Z. Guo, B. Shi, and N. Wang, A nonuniform lattice Boltzmann method based on domain decomposition, *Chinese J. Comput. Phys*. 18, 2001, 181-184

Table I Locations of Vortex of the Driven Cavity Flow

Re		Primary	First (T)	First (BL)	First (BR)	Second (BL)	Second (BR)
5000	a	0.5117, 0.5352	0.0625, 0.9102	0.0703, 0.1367	0.8086, 0.0742	0.0117, 0.0078	0.9805, 0.0195
	b	0.5176, 0.5373	0.0667, 0.9059	0.0784, 0.1373	0.8078, 0.0745	–	–
	c	0.5156, 0.5352	0.0625, 0.9063	0.0742, 0.1328	0.8086, 0.0742	0.0039, 0.0039	0.9961, 0.0742
7500	a	0.5117, 0.5322	0.0664, 0.9141	0.0645, 0.1504	0.7813, 0.0625	0.0117, 0.0117	0.9492, 0.0430
	b	0.5176, 0.5333	0.0706, 0.9098	0.0706, 0.1529	0.7922, 0.0667	–	–
	c	0.5156, 0.5352	0.0664, 0.9102	0.0664, 0.1484	0.7930, 0.0664	0.0078, 0.0039	0.9961, 0.0742
10000	a	0.5117, 0.5333	0.0703, 0.9141	0.0586, 0.1641	0.7656, 0.0586	0.0156, 0.0195	0.9336, 0.0625
	c	0.5117, 0.5313	0.0703, 0.9102	0.0625, 0.1563	0.7813, 0.0625	0.0117, 0.0117	0.9492, 0.0625
15000	c	0.5117, 0.5313	0.0781, 0.9141	0.0547, 0.1992	0.7227, 0.0391	–	0.9219, 0.0781
20000	c	0.5117, 0.5273	0.0820, 0.9102	0.0703, 0.1758	0.7109, 0.0391	–	0.8672, 0.0742

Note: The letters T, B, L, and R denote top, bottom, left, and right, respectively; a. Ghia *et al.* [10]; b, Hou and Zou [2]; c, present work.

Parallel GMRES for Solution of Reynolds Equation

Huang Chenxu Chen Xianqiao
Wuhan University of Technology, Wuhan, China 430063

ABSTRACT

The parallel GMRES is described in detail and its parallel algorithm program is developed, which is applied for the numerical solution of the Reynolds hydraulic lubrication equation.

Keywords: Parallel Algorithm of GMRES; Reynolds Equation; Algorithm of Arnoldi.

1. INTRODUCTION

To obtain the numerical solution of Reynolds is the core of hydraulic lubrication problem. Traditional numerical method of solving Reynolds equation is the finite difference method or finite element method [1][2]. Literature [3] discussed the application of GMRES on Solving Reynolds equation and showed that result is satisfactory. However, Those methods are all based on the series algorithms. Applying parallel GRMES for solving Reynolds equation isn't occurred yet. The parallel GMRES applied on algebraic equations after discretization of Reynolds equation is discussed in this paper, and the corresponding program is developed. The example shows that the resultant is satisfactory.

2. REYNOLDS EQUATION

Consider a non-stationary elastic flow problem, its basic equation

$$\frac{\partial}{\partial x}\left(\frac{\rho h^3}{\eta}\cdot\frac{\partial P}{\partial x}\right) = 12\,\mu\,\frac{\partial(\rho h)}{\partial x} + 12\,\frac{\partial(\rho h)}{\partial t} \qquad (1)$$

where $P(x,t)$——pressure of oil film
$h(x,t)$——thickness of oil file
$\rho(x,t)$——density
$\eta(x,t)$——viscosity of lubricating oil
$u(t)$——suck speed
t——time
x——coordinates

Boundary conditions:

$P(x_1,t)=0$ for miler x_1

$$P(x_n,t) = \frac{\partial}{\partial x}P(x,t)\Big|_{x=x_n} = 0$$

where x_1 ——inlet coordinate, determined by oil supply case;
x_n ——outlet coordinates in Reynolds boundary condition, to be determined in solving process

Geometry equation

$$h(x,t) = h_0(t) + \frac{x^2}{2R(t)} + \delta(x,t) \qquad (2)$$

where $R(t)$——equivalent radius of curve
$h_0(t)$——central thickness of film
$\delta_0(t)$——elastic deformation

Elastic deformation equation

$$\delta(x,t) = \frac{1}{XE'(t)}\int_{x_1}^{x_n} P(\xi,t)\ln|\xi - x|d\xi + c(t) \qquad (3)$$

Where $EX(t)$ ——equivalent elastic modulus
$C(t)$ —— isn't correspondence with x, t be determined

Load balance equation

$$\omega(t) = \int_{x_1}^{x_n} P(x,t)dx \qquad (4)$$

where $\omega(t)$ ——support force of oil film
viscosity ——Pressure relation

$$\eta(x,t) = \eta_0 exp[a\ p(x,t)] \qquad (5)$$

where η_0 —— the viscosity of lubrication oil under atmospheric pressure
α ——index of viscosity

Density-Pressure relation

$$\rho(x,t)/\rho_0 = B_T/(B_T - P(x,t))$$

Or

$$\rho(x,t)\rho_0 = 1 + 0.6\rho(x,t)/[1 + 1.7P(x,t)] \qquad (6)$$

where ρ_0 —— the density of lubrication oil under atmospheric pressure
B_T——Secant Bull modulus

Non-stationary Parameter——Time relation

(1)Cyclic $V(t)=v(t+t_p)$
(2)Non-cyclic $V(t)=f(t)$

where $V(t)$——Non-stationary parameter, for example, $u(t)$, $w(t)$ and so on.
t_p ——cycle

3. DISCRETE AND ITERATION SOLUTION

Substituting (5) into (1), then

$$\frac{\partial}{\partial x}(\frac{\rho h^3}{e^{ap}}\frac{\partial P}{\partial x}) = 12\eta_0 u\frac{\partial(\rho h)}{\partial x} + 12\eta_0\frac{\partial(\rho h)}{\partial t} \quad (7)$$

Substituting simplified pressure Q(*x*,*t*) into *P*(*x*,*t*) in (7) as unknown in order to decrease the difficult in numerical calculation, the premise condition is let *Q*(*x*,*t*) changes slower than that of *P*(*x*,*t*). Introducing Q(*x*,*t*), let

$$\frac{dQ}{dx} = \frac{1}{e^{ap}}\frac{dP}{dx}$$

integrate it and let $N\to\Gamma\infty$, then $Q=P=0$, so that

$$Q(x,t) = \frac{1}{a}(1-e^{-ap(x,t)})$$

Substituting above into (7), get

$$\frac{\partial}{\partial x}(th^3\frac{\partial Q}{\partial x}) = 12\eta_0 u\frac{\partial(\rho h)}{\partial x} + 12\eta_0\frac{\partial(\rho h)}{\partial t} \quad (8)$$

Discrete (8) it can be got that:

(1)
$$\frac{\partial(\rho h)}{\partial t} = \rho\frac{\partial h}{\partial t} + h\frac{\partial\rho}{\partial t} \approx \rho_{ij}\frac{h_{ij}-h_{ij-1}}{\Delta t} + h_{ij}\frac{\rho_{ij}-\rho_{ij-1}}{\Delta t}$$

(2)
$$\frac{\partial(\rho h)}{\partial x} = \rho\frac{\partial h}{\partial x} + \frac{\partial\rho}{\partial x} \approx \rho_{ij}\frac{h_{ij}-h_{i-1j}}{\Delta x} + h_{ij}\frac{\rho_{ij}-\rho_{i-1j}}{\Delta x}$$

(3)
$$\frac{\partial}{\partial x}(\rho h^3\frac{\partial Q}{\partial x}) \approx [\rho_{i+1,}h^3_{i+1,j}(Q_{i+1,k}-Q_{ij}) - \rho_{ij}h^3(Q_{ij}-Q_{i+1,j})]/\Delta x^2$$

where $t_j = t_{j-1}+\Delta t$, $h_{i,j-1} = h(x_i,t_{j-1})$, $\rho_{ij-1} = \rho(x_i,t_{j-1})$ are the thickness and density of film at last moment, as known.

Substituting (1), (2) and (3) into (8), we get a non-Linear equations. The following iteration algorithm can be applied in practices solving:

1) Suppose Layer $t=t_k$ is obtained.
2) In layer $t=t_{k+1}$ take arbitrarily value

$$P^{(0)}_{i,k+1}, i = 1, 2, \cdots, n$$

3) Calculate $h(x_i,t_{k+1})$, $\delta(x_i,t_{k+1})$, $w(t_{k+1})$, $\eta(x_i,t_{k+1})$, $\rho(x_i,t_{k+1})$
4) Substituting (1),(2) into Reynolds equation in order to get the approximate value $P^{(l)}_{i,k+1}$ of $P_{i,k+1}$
5) Calculate $\left\|P^{(0)}_{i,k+1} - P^{(l-1)}_{i,k+1}\right\| = e$, if $e > \varepsilon$, and ε is an accuracy given preliminary then turn to step 3, until get the satisfactory accuracy.

Note that, in every iteration of large sparse equations, if using traditional method, the consume and storage are all big, so the following GMRES algorithm is introduced.

4. GMRES ALGOTITNM

The GMRES algorithm is a kind of numerical method for solving asymmetry large sparse equations developed in the nineties, Its' as follow:

1) $\forall\ x_0\in R^n$ Calculate $r_0=b-Ax_0$, $v_1 = r_0/\|r_0\|$
2) Selecting the proper *m*, solve for $\{v_1, v_2, \cdots, v_m\}$ and $\overline{H}_m$ by Arnoldi method.
3) Minimize $\left\|\beta_{\mathcal{V}}^{1} - \widetilde{H}_m y\right\|$, get y_m
4) $x_m = x_0 + V_m y_m$, $V_m=\{v_1,v_2,\cdots,v_m\}$
5) Calculate $\|r_m\| = \|b-Ax_m\|$, if it is satisfactory stop.
6) $x=x_m$, $v_1=r_m/\|r_m\|$, $r_0=b-Ax_0$, turn to 2). Where the Calculation method of $v_1, v_2, \cdots, v_m$ is step 1). That is:

$$v_0 = r_0/\|r_0\|, \widetilde{v}_2 = Av_1 - h_{11}v_1$$

Therefore

$$v_2 = v_2/\|\widetilde{v}_2\|, h_{11} = (Av_1, v_2), \cdots$$

and

$$\widetilde{v}_{k+1} = Av_k - (\sum_{j=1}^{R} h_{ij}v_j), v_{k+1} = \widetilde{v}_{k+1}/\|\widetilde{v}_{k+1}\|$$
$$h_{k+1,k} = \|\widetilde{v}_{k+1}\|, v_{k+1,k} = \widetilde{v}_{k+1}/h_{k+1,k}$$
$$h_{i,k} = (Av_k, v_i) = 1,2,\cdots,k$$

H_m is the upper Hessenberg matrix

$$H_m = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{m-11} & h_{m1} \\ h_{21} & h_{22} & \cdots & h_{m-12} & h_{m2} \\ & h_{32} & \cdots & h_{m-13} & h_{m3} \\ & \cdots & & \cdots & \cdots \\ & & & h_{mm-1} & h_{mm} \end{bmatrix}$$

$$\overline{H} = \begin{bmatrix} H_m \\ h_{mH,m}e_m^T \end{bmatrix}, e_m^T = (0,0,\cdots 0,1)_m$$

The basic idea of GMRES algorithm is founded on the principle of Galerkin, its mains point is that when solving an algebra equations with bigger dimension by solving one in lower dimension to get a solution with designed accuracy which is satisfactory. This method for solving equations with higher dimension is not only increase the solving speed, but also decrease the storage space in solving process in determined degree.

The GRMES algorithm is introduced from the principle of Galerkin Consider solution for equation

$$AX=b \tag{9}$$

where A is non singular large sparse asymmetry matrix,
$b\in R^n$ is a designed vector, given.
the norm $\|\cdot\|$ —— α -norm, in following

Note R_m and Lm are two subspace with m dimension which are span by $\{v_i\}^m_{i=1}$ and$\{w_i\}^m_{i=1}$ take arbitrary $X_0\in R^n$ as a vector, *Let* $X=X_0+Z$, then formula (9) is equivalent to

$$AZ=r_0 \tag{10}$$

where $r_0=b-AX_0$

Solution for (10): Looking for the approximate solution Z_m of (10) in subspace K_m which makes the residue vector r_0-AZ_m is perpendicular to all vectors in subspace L_m. That is $Z_m\in K_m$.

$$(r_0-AZ_m,w)=0,\forall\, w\in L_m \tag{11}$$

Note $\{v_i\}^m_{i=1}$, $\{w_i\}^m_{i=1}$ are the basis of K_m and L_m. for (11)

$$(AZ_m,w)=(r_0,w) \tag{12}$$

Let $W_m=[w_1, w_2, \cdots, w_m]$, $V_m=[v_1, v_2, \cdots, v_m]$, then

$$W^T_mAZ_m=W^T_mr_0 \tag{13}$$

and $Z_m=V_my_m$, when $W^T_m\ AV_m$ is non-singular, the approximate solution can be obtained

$$Z_m=V_m(W^T_mAV_m)^{-1}W^T_mr_0 \tag{14}$$

In practical calculation it is difficult to assert whether the $W^T_m A V_m$ is non-singular. If using formula (14) and $W^T_mAV_m$ is singular, this leads to stop calculation.

If let $K_m=L_m=span\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$, then $W_m=V_m$ and it is easy to prove by mathematic inductive method that $\{v_i\}^m_{i=1}$ is an orthonormal base in span $\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$, and v_{m+1} is orthogonal to span $\{r_0, Ar_0, \cdots, A^{m-1}r_0\}$. Meanwhile the process of solution for (10) is named as Arnoldi algorithm process, that is

1) take $v_1=r_0/\|r_0\|$
2) k=1,2,$\cdots$, m, calculate

$$\tilde{v}_{k+1}=Av_k-\sum^{k}h_{i,k}v_i$$

$$h_{i,k}=(Av_k,v_i)$$

$$h_{k+1,k}=\|\tilde{v}_{k+1}\|$$

$$v_{k+1}=\tilde{v}_{k+1}/h_{k+1,k}$$

The matrix operating formula of Arnoldi algorithm process is

$$AV_m=V_mH_m+h_{m+1,m}V_{m+1}e^T_m \tag{15}$$

or

$$AV_m=V_{m+1}\overline{H}_m \tag{16}$$

Where H_m is Hessenberg matrix,

$$\overline{H}_m=\begin{bmatrix}H_m\\ h_{m+1,m}e^T_m\end{bmatrix}$$

Left multiplying the ends of (16) by V^T_m, note orthogonality, then we have

$$V^T_mAV_m=H_m$$

and $V^T_m=\beta e_i(\beta=\|r_0\|, e_i=(1,0,\cdots,0)^T)$
in that time,(14) becomes

$$H_my_m=\beta e_1 \tag{17}$$

When H_m is non-singular, it isn't difficult to solve for (7),so that we can get easy and approximate solution of Z_m, This is said as the Arnoldi algorithm Principle. if H_m is a singular matrix, the algorithm leads to be stopped in suspended.

If taking $K_m=span\ \{r_0, Ar_0, \cdots, A^{m-1}r_0\}$, L_m=span $\{Ar_0, A^2r_0, \cdots, A^mr_0\}$, or simplified $L_m=AK_m$, applying the Arnoldi process, get an orthonormal base of K_m: $\{v_i\}^m_{i=1}$, using (16),ge

Because $V^T_{m+1}\ V_{m+1}=I_m$, so that

$$\|r_0-AZ\|=\|\beta e_1-\overline{H}_my\|$$

$$\|r_0-AZ\|=\|r_0-AV_my\|=\|r_0-v_{m+1}\overline{H}_my\|$$

$$=\|V_{m+1}(\beta e,H_my)\| \tag{18}$$

(19)

Therefore $\|r_0-AZ\|=\min$ is equivalent to $\|\beta e_1-\overline{H}_my\|=\min$ in K_m.

For (9), take arbitrarily $x_0\in R^n$, $x=x_0+Z$, get (10) for fixed integer m>0, $K_m=span\ \{r_0, Ar_0\cdots, A^{m-1}r_0\}$, $L_m=AK_m$, the solution Z_m is obtained by the principle of Galerkin, which leads the 2-norm of the residual vector $r_m=r_0-AZ_m$ to minimize in the vector of all K_m. This is true inverse. The detail can be referred to literature [4]

The algorithm founded on this conclusion is named as General Minimal Residual Algorithm (GMRES).

5. PARALLEL GAMRES ALGOTITHM

Consider the solving region of (8) is Ω ,separating Ω into S subspace Ω_1, Ω_2, $\cdots$, Ω_S, $\Omega i\cap\Omega j=\phi$, $i\neq j$ the virtual boundary of Ωi is Γ_i, i=1,2,$\cdots$, s, suppose the layer t_k is gotten already, the practical algorithm to solve $t=t_k+1$ is following

(1) Predict the value of Γ_2, Γ_2, $\cdots$, Γ_{s-1}.

(2) Calculate P_{ij} in every Ω_i, $1\leqslant i\leqslant s$ according to the preceding algorithm, and let P_{ij} satisfy the designed accuracy in their subspaces.

(3) $\Gamma 2$, $\Gamma 3$, $\cdots$, $\Gamma s-1$ are modified, and calculate the error e_b between earlier of late virtual

boundaries two times, if $e_b < \varepsilon$, *stop*, unless turn to (2)

(4) Output and print the Calculation result.

6. EXAMPLE

According to the data supplied by[2] consider pure squeeze, the parameters are

η_0=0.048 (P_a, s)
α =2.06×10^{-3}(m^2 / N)
R=7 mm, roller radius
E′=1.21×1011 (N/m2), elastic modulus

The boundaries are

when $t=t_0$, $P(x,0)=0$, $H(0,0)=0.5$ (cm), $w_{(0)}=0$ (N/m), $t=t_1=0.1\times10^{-8}$ (s), $w(t_1)=2.8\times10^4$(N/m)

the convergence factor, pressure CP=5%,load CW=2%, the pressure distribution and relative thickness of file $H(x,t)-H_0$, *at t*, moment, the maximum central pressure $P_{max}\approx0.7$ (GPa) the corresponding maximum Hertz pressure $P_{Hmax}\approx0.28$ (GPa), and the minimum thickness of film H_{min}=0.6148 (mm).

The solving region is separated into Ω_1, Ω_2, Ω_3, Ω_4, t=0,1,2,3,4,5 the Calculation resultant is following

0.000e+000 1.396e − 002 1.727e − 002 2.027e−002 2.304e−002
2.555e − 002 2.778e − 002 2.972e − 002 3.916e−002 2.175e−002
3.210e − 002 3.333e − 002 3.425e − 002 3.444e−002 3.497e−002
3.516e − 002 3.502e − 002 3.493e − 002 3.437e−002 3.349e−002
3.322e − 002 3.195e − 002 3.040e − 002 2.816e−002 2.660e−002
7.485e−003 3.048e−003 0.000e+000

REFERENCES

[1] Wen Shizhu. The principle of friction. TsingHua University Press Inc.1990.China (in Chinese).
[2] Ren ning, Wen Shizhu. The direct iteration of a non-stationary elastic lubrication for linear osculation. A symposiums of the Fourth symposium of friction technology publish TsingHua University Press in 1985.27-32.China (in Chinese)
[3] Wu huapeng. The Application of GMRES Algorithm in calculation Reynolds Equation. Journal of Lubrication and Sealed, China 2~4,2001(in Chinese)
[4] Nachligal N M.GMRES Algorithm for No symmetric linear systems SIAM J Matrix Anal Appl.13 (1992). 796~825

Parallelization on Network of Workstations for a Model of Numerical Weather Prediction

Pham Hong UANG
Hanoi Institute of Mathematics
P.O. Box 631 Bo Ho, Hanoi – Vietnam
E-mail: phquang@hn.vnn.vn

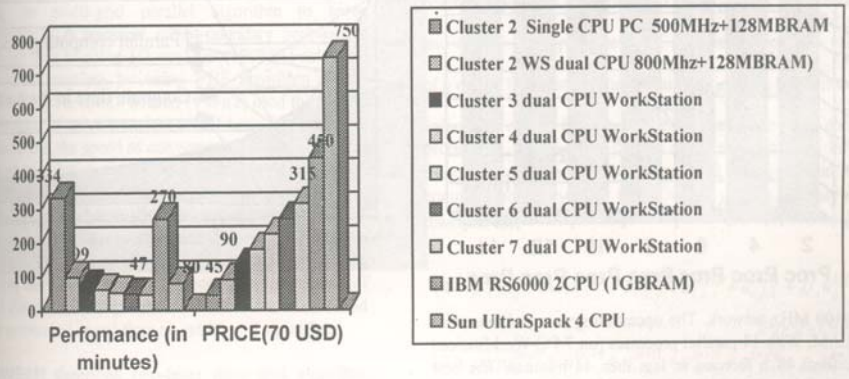
ABSTRACT

The report is on a parallelization of high-resolution model for distributed memory computing architecture. The source program is HRM, a regional numerical weather prediction program of the Deutscher Wetterdienst. It is written for shared memory vector computers or workstations, using OpenMP directives in standard FORTRAN 90. Our parallelization is written with MPI (Message Passing Interface) running on clustering dual Intel Pentium 800MHZ CPU computers. The comparison performance/price between share memory and distributed memory parallelization of this model is also analized.

Keyword: MPI Programming, Distributed Memory Models, Numerical Weather Prediction.

1. INTRODUCTION

Motivated by trends in numerical methods and high performance computing architectures the Deutscher Wetterdienst (DWD) has developed a new operational global numerical weather prediction (NWP) model that employs a grid point approach with an almost uniform icosahedral-hexagonal grid. On first December 1999 this new model has replaced the operational global model GM, derived from the spectral model of the European Centre for Medium Range Weather Forecasts (ECMWF), and the regional model for central Europe. It has been named GME and provides the meteorological database for many follow-up products and systems.



Forecast data from early run of the GME are sent via Internet to other national meteorological service (NMS). These data serve as initial and lateral boundary conditions for regional NWP models, which are based on either the high-resolution regional model or the nonhydrostatic LM. Currently, the following ten NMSs are receiving the GME data twice daily based on 00 and 12 UTC data out to 48(78) hours at 3-hourly intervals:

- Brazil (Directorate of Hydrography and Navigation);
- Brazil (Instituto National de Meteorologia) ;
- China (Guangzhou Regional Meteorological and Hydrological Service);
- Greece (National Meteorological and Hydrological Service);
- Israel (Israel Meteorological Service);
- Italy (Regional Service SMR-ARPA);
- Oman (National Meteorological Service, DGCAM);
- Poland (National Meteorological Service, IMGW);
- Romania (National Meteorological and Hydrological Service);
- Switzerland (National Meteorological Institute).

2. HIGH RESOLUTION MODEL IN VIETNAM

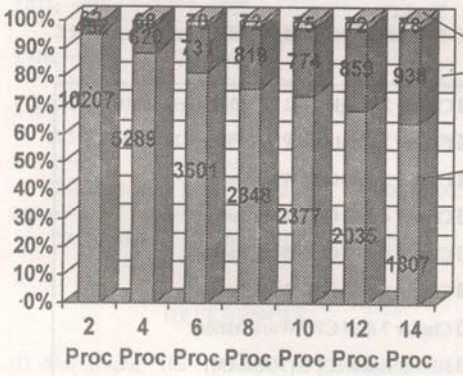
The regional high-resolution model (HRM) is a hydrostatic limited-area model based on the Europa-and Deutschland-Model of the Deutscher Wetterdienst. It uses the same physical parameterizations as the icosahedral-hexgonal gridpoint model GME of the DWD which provides lateral boundary data for the HRM. HRM uses an Arakawa C grid (horizontal, in rotated longitude/ latitude coordinates) and a

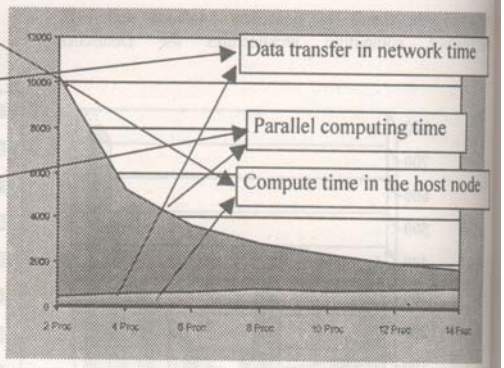
Lorenz grid (vertical, with a hybrid sigma/pressure based coordinate). The program is written for shared memory vector computers or workstations. There are ten large parallel regions in subroutine *progorg* each of these regions can be distributed to "nproc" processors. The paralleliszation uses standard OPEN-MP directives. According to DWD, the HRM has been tested successfully on the following computer systems: Cray C98, Cray J90, Cray SV1, SGI Origin 2000, and DEC, SUN and HP workstations.

In Vietnam, HRM has been tested on dual CPU IBM RS6000, 1GbRAM computer (at the National Center for Hydrological-Weather forecast) and on quad CPU Sun UltraSpack (at the Metrological Department – Hanoi National University). The resolutions are 25Km, 20 high levels at 150-second prediction interval. The prediction is positively evaluated, however the performance has been rather limited (it takes 270 minutes for the First and 120 minutes for the Second computer to perform 48 hours forecast against less then 60 minutes real processing requirement).

The parallelization concept with share memory architecture is quite easy to install: every subroutine will be distributed to one process to do computing a part of whole data domain. The data model is cut to "nproc" subdomains and the programmer will have to take care for data consistency in some neighbor rows between two regions. By choosing appropriate "nproc" to distribute computing to processors, the load balancing will be achieved.

We have parallelized the program using Message Passing Interface (MPI) for distributed memory based on Network of WorkStations (NOW). The NOW constructed from some dual Intel CPU 800 MHz, 128 MbRAM computers interconnected





by 10/100 MHz network. The operation system is Linux with MPI LAM. With 14 parallel processes (on 7 PC-WorkStation) we perform 48 h forecast in less then 44 minutes. The best performance can be achieved by optimizing the load balancing between the computing nodes and minimizing data transfer on the net. We build a point-to-point small part data exchange between neighbor computing nodes. All computing nodes have equivalent load factor, except only one node sometime plays the data in/out node.

The following is cost/performance diagram comparison between the 4 tested systems: cluster with single CPU PC 500MHZ, 128 MbRAM, cluster with dual CPU WorkStation Intel Pentium 800 MHZ, 128 MbRAM, dual CPU 450 MHZ mini computer IBM RS6000 and quad CPU Sun Ultra Spack. To achieve the 60-minute calculation for 48-hour prediction in Vietnam region we have to invest about 100.000 USD for Sun UltraSpack or IMB RS 6000, and 15.000 USD for clustering dual CPU network.

3. CALCULATION AND LATENCY TIME IN NOW

The following diagrams show the timing of processes calculation in our NOW. It proves that the calculation time is inversely proportional with number of computer node, while the data transfer and computing time in the host node (latency time of parallel system) does not increases very quickly. With HRM we can perform 48h forecast for 210x161x20 node grid in 30 minutes in 10 dual CPU, 100 Mbs network.

4. PARALLEL/DISTRIBUTED APPLICATIONS IN HANOI INSTITUTE OF MATHEMATICS

Beside the application reported in previous chapters, we developed a system based on clustering graphic and calculation computers simulating vehicle or aircraft driving on very large 3D landscape model. The system can realtime processes position calculating in 3D landscape model, realtime rendering on array of screens the vision...

We are going to build a new NOW with 10 dual Xeon 1.7Gh computers, 10 dual Pentium III 800 MHz, 2Gb network interconnection. This is supposed to be strongest parallel computing center in Vietnam.

REFERENCES

[1] Detlev Majewski, Dorte Liermann, Peter Prohl and Bodo Ritter, Michael Buchhold, Thomas Hanish, Gerhard Paul and Werner Wergen. The global icosahedral- hexagonal grid point model GME: Operational version and high resolution tests. DWD, Research and Development Division, Offenbach, Germany.

A Multi-grid Parallel Computing Model of Heat Transfer in Two-Layered Material

Wei Jianing[1] Zhang Sheheng[1] Guo Qingping[1]
Yakup Paker[2] Dennis Parkinson[2]
[1]Transportation University, Wuhan 430063, P. R. China
[2]Qeen Mary & West field College, University of London. E1 4NS U.K.
She77mei@public.wh.bh.cn paker@dcs.qmw.ac.uk

ABSTRACT

This article discusses the multi-grid parallel heat transfer model of the object combined with tow different materials. By using of the temperature continuing condition, we obtain the initial and boundary condition problem, and a group of the difference equation solved by the VBF algorithm. The numerical data show that our algorithm has high speedup, and less calculation time than no-VBF algorithm.

Keywords: Algorithm, Multi-grid Parallel Computing, Difference Equation

1. INTRODUTION

In practical, there is an object combined with two different materials. Such as ceramic and metal are combined into a combining material, which widely be used in industry. Heat transfer problem of combining material is more complex than the one of no-combining material.

By using the multi-grid parallel ·algorithm to solve two-dimension (2D) transient initial-boundary problem is more difficult than to solve 1D one. Guo(1998)[1] solve the boundary-initial problem by using VBF algorithm (VBF: Virtual Boundary Forecast). Zhang(1998)[2] used the simple iterative method to solve transient initial-boundary problem, in his calculation, the speed of convergence is very slow, and spend lot of calculation time to obtain convergence results. By using domain decomposition parallel iterative algorithm, Guo (1998)[3] solved one dimension Ceramic/Metal heat transient equation, the speedup can be reached 16, the convergence speed was much faster than paper[2]. Xu(1996)[4] discussed the domain decomposition method of multi-grid distributed computer, and pointed out this method had faster converged speed than no-multi-grid algorithm.

Wang (1992)[5] described two-lever multi-grid algorithm and full multi-grid algorithm, and given some examples in the fluid dynamics. The communication always takes place when multi-grid parallel algorithm is used, that reduce the ratio of speedup calculation efficiency.
The paper will use multi-grid VBF algorithm to solve 2D heat transient combining initial-boundary problem

* Research supported by the UK Royal Society joint project (the Royal Society Q724), the Natural Science Foundation of China (NSFC Grant No. 69773021) and the Natural Science Foundation of Hubei Province (NSFHP 2000 J153).

2. DIFFERENCE EQUATION MULTIGRID ALGORITHM

In this paper, we suppose here Γ is boundary of domain W, k_1 and k_2 are heat conductive coefficient of material A and material B respectively, u is temperature. Consider two-dimension heat transient combining initial boundary problem as following:

$$\frac{\partial u}{\partial t} = k_A\left(\frac{\partial^2 u}{\partial^2 x}+\frac{\partial^2 u}{\partial^2 y}\right) \quad (x,\ y)\in\Omega_A \quad t>0$$

$$\frac{\partial u}{\partial t} = k_B\left(\frac{\partial^2 u}{\partial^2 x}+\frac{\partial^2 u}{\partial^2 y}\right) \quad (x,\ y)\in\Omega_B \quad t>0$$

$$u|_\Gamma = f(x,\ y,\ u,\ t) \quad (x,\ y)\in\Gamma$$

$$u|_{\Gamma ab} = u|_{\Gamma ba}$$

$$u(x,\ y,\ 0) = \psi(x,\ y) \quad (x,\ y)\in\Omega \qquad (1)$$

Here $\Omega = \Omega_A \cap \Omega_B$, Γ is boundary of domain Ω, k_A and k_B are the heat conductive coefficient of material A and B respectively. Γ_{ab} is the common boundary of the domain Ω_A and Ω_B, u is temperature of the material, $u|_{\Gamma ba}$ is the value of u on the boundary of the domain Ω_A and $u|_{\Gamma ab}$ is the value of u on the boundary of the domain Ω_B. From above problem, k_A and k_B may be two difference constants, but u must be continued in the domain Ω, especially on the common boundary Γ_{ab}. Let $u_{ij}=u(x_i,\ y_j)$, Eq.(1) may be rewritten as:

$$(4+\rho_A)u_{ij}^{(n+1)} = u_{i-1,\ j}^{(n+1)} + u_{i+1,\ j}^{(n+1)} + u_{i,\ j-1}^{(n+1)} + u_{i,\ j+1}^{(n+1)} + \rho_A u_{ij}^{(n)}$$
$$(x_i,\ y_j)\in\Omega_A$$

$$(4+\rho_B)u_{ij}^{(n+1)} = u_{i-1,\ j}^{(n+1)} + u_{i+1,\ j}^{(n+1)} + u_{i,\ j-1}^{(n+1)} + u_{i,\ j+1}^{(n+1)} + \rho_B u_{ij}^{(n)}$$
$$(x_i,\ y_j)\in\Omega_B$$

Here

$$\rho_A = \frac{h^2}{k_A \Delta t} \qquad \rho_B = \frac{h^2}{k_B \Delta t} \qquad (2)$$

i=0, 1, 2, ···, M_x, j=0, 1, 2, ···, M_y, n=0, 1, 2, ···

h and Δt are step length of space and time t respectively, M_x, M_y are nods number in x and direction respectively. Suppose u^* is approach value of u and $u = u^* + \phi$, ϕ satisfy following formula

$$\varphi = k_s(\varphi_{xx} + \varphi_{yy}) + r$$

here

$$r = k_s(u^*_{xx} + u^*_{yy}) - u^*_t \qquad s = A, B$$

$$\varphi|_\Gamma = 0, \quad \varphi|_{\Gamma ab} = \varphi|_{\Gamma ba}, \quad \varphi(x, y, 0) = 0 \tag{3}$$

Its difference formula is:

$$(4+\rho_s)\varphi_{ij}^{(n+1)} = \varphi_{i-1,j}^{(n+1)} + \varphi_{i+1,j}^{(n+1)} + \varphi_{i,j-1}^{(n+1)} + \varphi_{i,j+1}^{(n+1)} + r_{ij}\frac{h^2}{\Delta t}$$

$$\varphi|_\Gamma = 0 \qquad s = A, B \tag{4}$$

$$\varphi(x_i, y_j, 0) = 0$$

In this paper, the main multi-grid calculate step of solving Eq.1) are following:

(a) By using initial function value $u_{ij}^{(n-1,0)}$, on the fine grid $\Omega^{(h)}$, calculate ρ_A and ρ_B;
(b) From Eq.2), using iterative method to obtain the approach value $u_{ij}^{(n-1,s)}$, in this paper iterative times is 2;
(c) Calculate residua r_{ij} on the fine grid $\Omega^{(h)}$, and restrict it to the coarse grid $\Omega^{(H)}$;
(d) On the coarse grid $\Omega^{(H)}$, using iterative method get value of $\phi_{ij}^{(n-1)}$ from Eq.4)
(e) By using linear interpolate method, calculate $u_{ij}^{(n-1)} = u_{ij}^{(n-1,*)} + \phi_{ij}^{(n-1)}$ on the fine grid $\Omega^{(h)}$.
(f) By using smooth method, calculate $u_{ij}^{(n-1,1)}$ on the fine grid $\Omega^{(h)}$

3. MULTI-GRID PARALLEL ALGORITHM OF VBF

In the paper [1], we have defined VBF, which means the virtual boundary forecast algorithm. The purpose of it is to reduce communication between the computers. We use domain decomposition parallel algorithm, obtain sub-domain Ω_p, p=1,2,…, P. Suppose Γ_i is virtual boundary of Ω_i that exclude actual boundary Γ. The whole virtual boundary may be written as:

$$\Gamma_V = \Gamma_1 + \Gamma_2 + \cdots + \Gamma_p$$

When we use the parallel calculate method to solve Eq.1), the communication always takes place on the virtual boundary. It reduce the speedup and efficiency. This section will give two-dimension virtual boundary forecast method to reduce times of communication on virtual boundary Γv.

Suppose $u_k = u_k(x)$ are the numerical value of function u on the virtual boundary when iterative times is k'th, and $x \in \Gamma v$ is a point of space R^2, then $\{u_k\}$ is a sequence of points, k=1,2,… m we construct a subspace of R^m

$$R^m = \{u_1, u_2, \ldots, u_m\}$$

Assume Q is a mapping R^m to R

$$W_{m+1} = Q(Y) \qquad Y \in R^m$$

This paper takes W_{m+1} as virtual boundary forecast value of function u on m+1)'th iteration. Map G has many kinds of the formula, this paper use the limit value formula [7]. The forecast procedures are:

(1) Decomposition domain Ω into sub-domain Ω_k k=1,2,…,P, and

$$V_K = U(X_K) \qquad X_K = \Omega \cap \Omega_K^{(h)}$$

Here X_k is a set of points belongs to $\Omega_k^{(h)}$
(2) Given initial iterative value $V_k^{(0)}$, relative virtual boundary value on point x is b_0, error bounds ε; fine grid $\Omega^{(h)}$ and coarse grid $\Omega^{(H)}$, integer N_{min}, initial iterative number s=0;
(3) Using non-linear multi-grid parallel method describe at above section, from Eq.2) calculate $V_k^{(1)}$ and $V_k^{(2)}$, relative virtual boundary value on point x are b_1 and b_2 respectively:
(4) s=s+1; Forecast W_3 by following Limited value formula [3]

$$w_3 = b_2 - \frac{a^2}{4b}$$

here a=0.5(b_2−b_0), b=0.5(b_2−2b_1+b_0),
(5) Take W_3 as the virtual boundary value, calculate the function value $V_k^{(3)}$ on the sub-domain Ω_k, k=1,2,… P, in this step needn't communicate between sub-domain, multi-grid calculation only take place at sub-domain;
(6) Using Eq.2), calculate $V_k^{(4)}$ on whole domain Ω, in this step we didn't use multi-grid method, only use smooth iterate method, but communicate must take place between sub-domain; relative virtual boundary value are b_4; In this step there isn't communication on the coarse grid $\Omega^{(H)}$, because we don't calculate modified value ϕ in Eq.3)
(7) If iterative number s<N_{min}, rewrite $b_4 \to b_2$, $W_3 \to b_1$, $b_2 \to b_0$, go to step (3); Otherwise do next step;
(8) If $\|V_k^{(4)} - V_k^{(3)}\| < \varepsilon$, go to step (9), otherwise go to step (3);
(9) End iterative calculation on sub-domain Ω_k, output numerical data of u, and end communication between Ω_k and its adjoin sub-domain;
(10) If function u is converged on all sub-domains, we end calculation on whole domain Ω.

4. TWO MATETRIALS HEAT TRANSIENT EQUATION

In this section we consider an example of the two material heat transient problem as following:

$$\frac{\partial u}{\partial t} = k_A\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right) \quad 0<x<4, \quad 0<y<5 \quad t>0$$

$$\frac{\partial u}{\partial t} = k_B\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right) \quad 1<x<5, \quad 0<y<5 \quad t>0$$

$$u(x, y, 0) = 0 \qquad u(0, y, t) = \sin(2\pi)\sin(\pi y) \qquad k_A = 1.0$$

$$u(5, y, t) = 0 \qquad u(x, 0, t) = 0 \qquad u(x, 5, t) = 0 \qquad k_B = 4.0$$

$$u(4-0, y, t) = u(4+0, y, t)$$

From above, we know that the heat conductive coefficient of material A is K_A=1.0, and that the heat conductive coefficient of material B is K_A = 4.0 which is greater that K_A So that material B has better conductive than material A. Eq (5)

may be rewritten as

$$u_t = k_s(u_{xx} + u_{yy}) \qquad s = A, B$$

Difference formula of Eq.5)may be written as:

$$(4+\rho_s)u_{ij}^{(n+1)} = u_{i-1,j}^{(n+1)} + u_{i,j-1}^{(n+1)} + u_{i,j+1}^{(n+1)} + \rho_s u_{ij}^{(n)}$$

$$i,j = 0,1,2,\ldots,M;\ n = 0,1,2,\ldots$$

$$u_{0j}^{(n)} = \sin(2\pi t^{(n)})\sin(2\pi y_j) \qquad u_{Mj}^{(n)} = 0$$

$$u_{ij}^{(0)} = 0 \qquad u_{i0}^{(0)} = 0 \qquad u_{iM}^{(0)} = 0$$

$$u(4-0, y_j, t^{(n)}) = u(4+0, y_j, t^{(n)})$$

$$\rho_A = h^2/\Delta t \quad \rho_B = 0.25h^2/\Delta t$$

h and Δt are step length of space and time t respectively. From Eq.3),Suppose $u^*$is a approach value of u and $u=u^*+\phi$, ϕsatisfy following formula

$$\varphi_t = k_s(\varphi_{xx} + \varphi_{yy}) + r$$

here $r = k_s(u^*_{xx} + u^*_{yy}) - u^*_t$

Its difference formula is:

$$(4+\rho_s)\varphi_{ij}^{(n+1)} = \varphi_{i-1,j}^{(n+1)} + \varphi_{i+1,j}^{(n+1)} + \varphi_{i,j-1}^{(n+1)} + \varphi_{i,j+1}^{(n+1)} + r_{ij}\frac{h^2}{\Delta t}$$

$$\varphi(0, y_j, t)=0 \quad \varphi(5, y_j, t)=0 \qquad \varphi(x_i, 0, t)=0$$

$$\varphi(x_i, 5, t)=0 \quad \varphi(x_i, y_j, t)=0 \quad \varphi(4+0, y_j, t)=\varphi(4-0, y_j, t)$$

The Table 1 shows the ratio of speedup of this example in the table; P is the number of using computers. The number of points is 40000 in fine grid.

Table 1 Ratio of Speedup

P	1	2	4	5	10	24
Speed up	1	1.88	3.62	4.33	6.85	11.30
Sp[3]	1	1.03	0.95	0.86	0.53	

Table 2 shows the calculation time ratio of our algorithm with no-forecast algorithm. The number of points in the fine grid is 14400. The results in Table 2 show that the forecast algorithm is much better than on-forecast algorithm, the calculation time ratio may reach to 7.11 when the number of computers is p=6.

Table 2 Time Ratio

P	2	3	4	5	6
Time Ratio	2.95	4.00	4.76	5.64	7.11

5.CONCLUSIONS

VBF is an efficient algorithm of solving two material heat transfer problem, that often take place in engineer field. The most character is that function u is continue on whole domain and may be not continue on the common boundary of two materials. So that we have two control equation in the initial-boundary problem which is different with the one in paper [3]. The difference equations are obtained independently in two domains, and composed with the continue condition The VBF algorithm is also used independently in each domain.

Because the heat conductive coefficient is constant in each domain, only has gap on the common boundary, so that the initial-boundary problem is linear in each domain.
Our results show the VBF algorithm is efficient to solve two material heat transfer problem.

REFERENCES

[1]Qingping Guo et, Optimize Algorthm of Multi-grid Parallel with Virtual Boundary Forecast, Jour. Nume. And Comp. 2(2000).

[2]Zhang Shesheng, Wang Xianfu. Coal Particles dispersion in atmosphere boundary layer. 1988, Jour. Wuhan Transportion University. 34-41

[3]Guo qingping, Yakup Paker, et.. Parallel computing using domain decomposition for cyclical temperatures in ceramic/metal composites. London conference. 1998. London.

[4]Xu Zhengquan. Domain decomposition method of multi-grid distributed computer. Journal of Numerical and Compution Vol 1.1996.1-5

[5]Wang Xianfu Numerical methods of fluid dynamics. Shanghai. 1992.231-318

[6]Zhang Shesheng. Xiao. Kaiyong Numerical method published by Wuhan transportation University.1998, 134-178

[7]Guo Qingping, Yakup Paker.et. Network Computing Performance Evaluation In PVM Programming Environment. London conference.1998.London.

[8]Fan Yinshuan. Advanced mathematics. Beijing, 1962,pp 211-234.

An Iterative Method for Indefinite Linear Systems of Algebraic Equations

Gang Xie
Institute of Computer Applications, CAEP, P.R.China.
Email: xieg@caep.ac.cn

ABSTRACT

By generalizing the CG, we obtain a kind of short-recursion iterative method for indefinite linear systems. Its computation complexity is less than the normalization method because it does one time fewer matrix-vector multiplications in each iterative step. Its storage complexity is less than GCR because it is of short recursion while the later is of long recursion. It converges in finite steps with a smoothly decreasing residual norm because it has got a minimal residual.

Keywords: Numerical Computational Methods, Indefinite Linear Systems, Short Recursion Iterative Method, Normalization Method, GCR.

1. INTRODUCTION

It is well known that the design of numerical methods for linear systems of equations is a basal work for scientific computing. For SPD linear systems, the CG method is quite simple and efficient, but the methods for non-symmetric linear systems, such as FOM, GMRES, BCG, CGS ,QMR and so on, are much more complicated and less efficient. To solve indefinite symmetric linear systems, of course we cannot use methods for SPD problems, but it is unnecessarily expensive to use methods for non-symmetric problems. Here we suggest a special method for indefinite symmetric linear systems. Taking advantage of the symmetry of the problem, the method is almost as simple as the CG. It is not only of short recurrence but has got a minimal residual as well. Because symmetry is much easier to validate than positive definiteness and indefinite symmetric problems are much more numerous than SPD problems, it is quite significant to put forward this method.

2. DEFINITION OF THE METHOD

To begin with, let us consider the iterative algorithm of the linear system

$$Ax = b$$

Where A is a $n \times n$ nonsingular matrix. Let x_k be the k th iterate and $r_k = b - Ax_k$ be the corresponding residual. Then the Krylov subspace is

$$K_k(A, r_0) = span\{r_0, Ar_0, \ldots, A^k r_0\} \qquad (k \le n).$$

The algorithm is defined as follows.

Initial value: x_0 , r_0 , $p_0 = r_0$

iteration:

$$x_{k+1} = x_k + \alpha_{k+1} p_k$$

$$r_{k+1} = r_k - \alpha_{k+1} A p_k \qquad (1)$$

Here the parameter α_{k+1} is defined by

$$(Ap_k)^T r_{k+1} = 0$$

Furthermore

$$p_{k+1} = r_{k+1} + \beta_{k+1} p_k \qquad (2)$$

Here the parameter β_{k+1} is defined by

$$(Ap_k)^T A p_{k+1} = 0$$

3. PROPERTY OF THE METHOD

Theorem 1. Let A be a $n \times n$ nonsingular symmetric matrix and $p_i \neq 0$ for $i < m$, then for $0 \le i < k < m$ we have

$$(Ap_i)^T r_k = 0, \qquad (3)$$

$$(Ar_i)^T r_k = 0, \qquad (4)$$

$$(Ap_i)^T A p_k = 0 \qquad (5)$$

and

$$span\{r_0, r_1 \cdots, r_k\} = span\{p_0, p_1, \cdots, p_k\} = K_k(A, r_0). \qquad (6)$$

Here the three subspaces are all of $k+1$ dimensions.

Proof. We do mathematics induction for k . It is easy to verify that the formulas in theorem1 are true when $k = 1$. Suppose the formulas are true for $k = j < m - 1$, let us consider the case $k = j + 1$.

(A) If $i = j$, by definition of the iterative method we have

$$(Ap_j)^T r_{j+1} = 0$$

and if $i < j$, by (1) and the induction hypothesis we have

$$(Ap_i)^T r_{j+1} = (Ap_i)^T (r_j - \alpha_{j+1} A p_j) = 0,$$

so (3) is true for $k = j + 1$.

(B) By the induction hypothesis we have

$$span\{r_0, r_1 \cdots, r_j\} = span\{p_0, p_1, \cdots, p_j\},$$

so

$$span\{Ar_0, Ar_1 \cdots, Ar_j\} = span\{Ap_0, Ap_1, \cdots, Ap_j\}$$

By what (A) has proved we get

$$r_{j+1} \perp span\{Ap_0, Ap_1, \cdots, Ap_j\},$$

so

$$r_{j+1} \perp span\{Ar_0, Ar_1 \cdots, Ar_j\}.$$

Hence (4) is true for $k = j+1$.

(C) If $i = j$, by definition of the iterative method we have

$$(Ap_j)^T Ap_{j+1} = 0.$$

If $i < j$, by (2), the symmetry of A and the induction hypothesis we have

$$\begin{aligned}(Ap_i)^T Ap_{j+1} &= (Ap_i)^T A(r_{j+1} + \beta_{j+1} p_j) \\ &= (Ap_i)^T Ar_{j+1} = r_{j+1}^T AAp_i\end{aligned} \tag{7}$$

and by (1) we have

$$Ap_i = 1/\alpha_{i+1}(r_i - r_{i+1}) \tag{8}$$

so by (7),(8) and what (B) has proved we get

$$\begin{aligned}(Ap_i)^T Ap_{j+1} &= 1/\alpha_{i+1} r_{j+1}^T A(r_i - r_{i+1}) \\ &= 1/\alpha_{i+1}\left((Ar_i)^T r_{j+1} - (Ar_{i+1})^T r_{j+1}\right) = 0\end{aligned}$$

Hence (5) is true for $k = j+1$.

(D). By the induction hypothesis we have

$$r_j, p_j \in K_j(A, r_0),$$

so

$$r_{j+1} = r_j - \alpha_{j+1} Ap_j \in K_{j+1}(A, r_0),$$

and so

$$p_{j+1} = r_{j+1} + \beta_{j+1} p_j \in K_{j+1}(A, r_0)$$

By what (C) has proved we get

$$p_i^T A^T Ap_{j+1} = (Ap_i)^T Ap_{j+1} = 0$$

Because $A^T A$ is SPD, it defines a norm. Hence p_{j+1} is orthogonal to the vector group $\{p_0, p_1, \cdots, p_j\}$ according to this norm. Hence by the induction hypothesis we see the vector group $\{p_0, p_1, \cdots, p_{j+1}\}$ is independent. Because the vector group $\{p_0, p_1, \cdots, p_{j+1}\}$ is expressible by the vector group $\{r_0, r_1, \cdots, r_{j+1}\}$, the later is also independent. Hence (6) is true for $k = j+1$. By now we have proved theorem 1.

Theorem 2. Let A be a $n \times n$ nonsingular symmetric matrix and $p_i \neq 0$ for $i < m$, then for $0 < k \leq m$ we have

$$r_k \perp Az \qquad \forall z \in K_{k-1}(A, r_0) \tag{9}$$

and

$$\|r_k\| = \min\{\|b - Ax\| : x \in x_0 + K_{k-1}(A, r_0)\} \tag{10}$$

Proof. Lets prove (9) and (10) respectively.

(E). For $0 < k < m$, it is easy to see by theorem 1 that

$$r_k \perp Az \qquad \forall z \in K_{k-1}(A, r_0).$$

Furthermore, by the definition of the iterative method we have

$$(Ap_{m-1})^T r_m = 0,$$

and for $i < m-1$, by (1),(3) and (5) we have

$$(Ap_i)^T r_m = (Ap_i)^T (r_{m-1} - \alpha_m Ap_{m-1}) = 0$$

so it follows from (6) that

$$r_m \perp Az \qquad \forall z \in K_{m-1}(A, r_0)$$

Hence (9) is true.

(F). By theorem 1 we have

$$r_k \in K_k(A, r_0)$$

We can prove by mathematics induction that r_k is expressible as

$$r_k = r_0 + Az_k$$

With some

$$z_k \in K_{k-1}(A, r_0),$$

So we have

$$z + z_k \in K_{k-1}(A, r_0) \quad \forall z \in K_{k-1}(A, r_0).$$

By what (E) has proved we get

$$r_k \perp A(z + z_k),$$

So

$$\begin{aligned}&\|r_0 - Az\|^2 \\ &= \|r_k - A(z + z_k)\|^2 \\ &= \|r_k\|^2 + \|A(z + z_k)\|^2,\end{aligned}$$

and so

$$\|r_k\| \leq \|r_0 - Az\| \qquad \forall z \in K_{k-1}(A, r_0).$$

It follows that

$$\begin{aligned}&\|r_k\| \\ &= \min\{\|r_0 - Az\| : z \in K_{k-1}(A, r_0)\} \\ &= \min\{\|b - Ax\| : x \in x_0 + K_{k-1}(A, r_0)\}\end{aligned}$$

By now, we have completed the proof..

Theorem 3. Let A be a $n \times n$ nonsingular symmetric matrix, $p_i \neq 0$ for $i < m$, $p_m = 0$ and $\alpha_m \neq 0$, then $r_m = 0$.

Proof. Because $p_m = 0$, by (2) we have

$$r_m = -\beta_m p_{m-1} \in span\{p_0, p_1, \cdots, p_{m-1}\},$$

and by (1) we have

$$Ap_{m-1} = 1/\alpha_m (r_{m-1} - r_m).$$

Hence it follows from (6) that

$$Ap_{m-1} \in span\{r_0, Ar_0, \cdots, A^{m-1} r_0\}. \quad (11)$$

Furthermore, by (6) we have

$$span\{p_0, p_1, \cdots, p_{m-2}\} = span\{r_0, Ar_0, \cdots, A^{m-2} r_0\},$$

so

$$Ap_i \in span\{r_0, Ar_0, \cdots, A^{m-1} r_0\} (i < m-1) \quad (12)$$

It follows from (6) that $A^{m-1} r_0$ is expressible by $\{p_0, p_1, \cdots, p_{m-1}\}$. Namely

$$A^{m-1} r_0 = \sum_{i=0}^{m-1} a_i p_i$$

for some constants $a_0, a_1, \cdots, a_{m-1}$,

so

$$A^m r_0 = \sum_{i=0}^{m-1} a_i Ap_i \quad (13)$$

It follows from (11), (12) and (13) that

$$A^m r_0 \in span\{r_0, Ar_0, \cdots, A^{m-1} r_0\}.$$

Hence

$$span\{r_0, Ar_0, \cdots, A^{m-1} r_0\} = span\{r_0, Ar_0, \cdots, A^{m-1} r_0\}.$$

By Cayley-Hamilton theorem we have

$$A^{-1} b \in x_0 + span\{r_0, Ar_0, \cdots, A^{n-1} r_0\},$$

so

$$A^{-1} b \in x_0 + span\{r_0, Ar_0, \cdots, A^{m-1} r_0\}$$

It follows from theorem 2 that

$$\|r_m\| = min\{\|b - Ax\| : x \in x_0 + span\{r_0, Ar_0, \cdots, A^{m-1} r_0\}\} = 0$$

Theorem 4. Let A be a $n \times n$ nonsingular symmetric matrix and $p_i \neq 0$ for $i < m$, then

$$\alpha_m = 0 \Leftrightarrow r_{m-1}^T A r_{m-1} = 0$$

Proof. From the definition of the iterative method we get

$$\alpha_m = (Ap_{m-1})^T r_{m-1} / (Ap_{m-1})^T Ap_{m-1}$$

So

$$\alpha_m = 0 \Leftrightarrow (Ap_{m-1})^T r_{m-1} = 0.$$

By theorem 1 we have

$$span\{r_0, r_1 \cdots, r_{m-1}\} = span\{p_0, p_1, \cdots, p_{m-1}\}$$

and

$$r_{m-1} \perp span\{Ar_0, Ar_1 \cdots, Ar_{m-2}\} = span\{Ap_0, Ap_1, \cdots, Ap_{m-2}\}.$$

It follows that

$$(Ar_{m-1})^T r_{m-1} = 0 \Leftrightarrow (Ap_{m-1})^T r_{m-1} = 0$$

Notice that A is symmetric, so

$$\alpha_m = 0 \Leftrightarrow r_{m-1}^T A r_{m-1} = 0.$$

4. NUMERICAL EXPERIMENT

To test the performance of the algorithm, we do some computations for a number of examples. We set the degree of the equations as $n = 2000$ and define the relative residual as $re = \|r_k\| / \|r_0\|$. In the following diagrams, the k-axis represents the number of iterations, the re-axis represents the relative residual and NDS represents our iterative method.

Example 1. This is a non-symmetric four diagonal system. The coefficient matrix is

$$a_{i,i} = i \qquad (1 \le i \le n)$$
$$a_{i,j+1} = 1 \qquad (1 \le i \le n-1)$$
$$a_{i,j-2} = 1 \qquad (3 \le i \le n)$$

with the other elements as 0. For the results of computation see diagram 1 and diagram 2.

Example 2. This is a symmetric nine diagonal system. The coefficient matrix is

$$a_{i,j} = 4 \qquad (1 \le i \le n)$$
$$a_{i,j+1} = 3.5 \qquad (1 \le i \le n-1)$$
$$a_{i,j-1} = 3.5 \qquad (2 \le i \le n)$$
$$a_{i,j-2} = 2. \qquad (3 \le i \le n)$$
$$a_{i,j+2} = 2 \qquad (1 \le i \le n-2)$$
$$a_{i,j+3} = 7 \qquad (1 \le i \le n-3)$$
$$a_{i,j-3} = 7 \qquad (4 \le i \le n)$$
$$a_{i,j+4} = 6 \qquad (1 \le i \le n-4)$$
$$a_{i,j-4} = 6 \qquad (5 \le i \le n)$$

with the other elements as 0.

For the results of computation see diagram 3 and diagram 4.

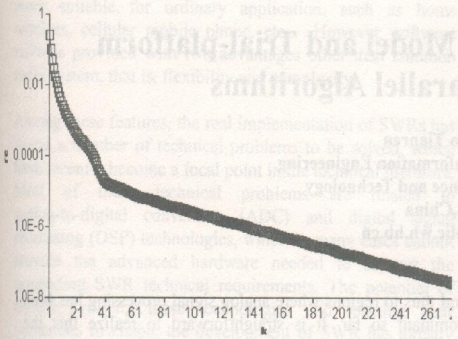


Diagram 1: NDS

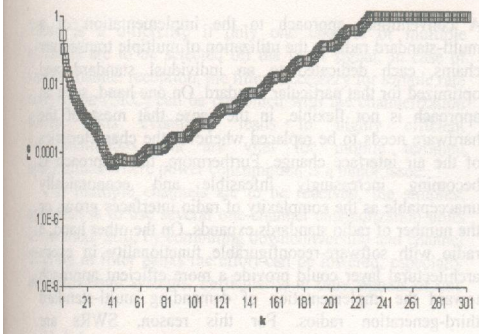


Diagram 2: CG does not converge.

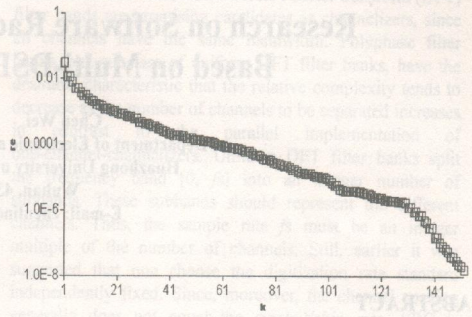


Diagram 3: NDS

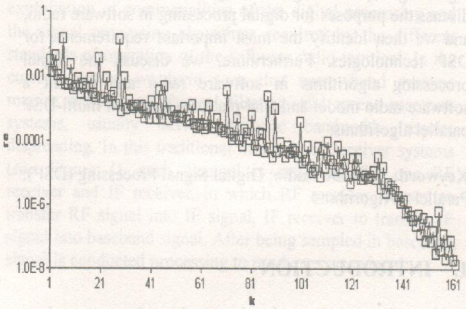


Diagram 4: CG

5. CONCLUSION

By generalizing the CG, we obtain a new short-recursion iterative method for indefinite linear systems. Its computation complexity is less than the normalization method because it does one time fewer matrix-vector multiplications in each iterative step. Its storage complexity is less than the GCR because it is of short recursion while the later is of long recursion. It converges in finite steps with a smoothly decreasing residual norm because it has got a minimal residual. Through the design, study and numerical experiment of the iterative method, we realize that residual smoothing techniques are absolutely unnecessary. The so-called smoothing is relative to a special norm. The residual of a method may be smooth and minimal for one norm, but not so for another norm. For example, the CG is smooth and minimal for the norm $\|x\|_A$, but not for the norm $\|x\|_{A^TA} = \|r\|_2$, this can be seen from the diagrams in this paper.

REFERENCES

[1] Kim-Chuan Toh., GMRES vs. Ideal GMRES, SIAM J. MATRIX ANAL. APPL. Vol.18, No.1, pp. 30-36,1997.

[2] Jane Cullum and Anne Greenbaum , Relations Between Galerkin and Norm-Minimizing Iterative Methods for Solving Linear Systems, SIAM J. MATRIX ANAL. APPL. Vol. 17, No. 2,pp. 223-247,1996.

[3] Anne Greenbaum, Vlastimil Pták, Zdenuek Strakous , Any Nonincreasing Convergence Curve is Possible for GMRES, SIAM J. MATRIX ANAL. APPL. Vol. 17, No.3, pp. 465-469,1996.

Research on Software Radio Model and Trial-platform Based on Multi-DSP Parallel Algorithms

Chen Wei Yao Tianren
Department of Electronic and Information Engineering
Huazhong University of Science and Technology
Wuhan, 430074,China
E-mail: chenlin@public.wh.hb.cn

ABSTRACT

Software radios are emerging as platforms for multiband multimode radio communication systems. At the same time, the need for software radios also raises a number of technical challenges, including analog-to-digital conversion and digital signal processing technologies. In this article, we first discuss the purposes for digital processing in software radio, and we then identify the most important requirements for DSP technologies. Furthermore, we discuss the signal processing algorithms in software radio and present a software radio model and trial-platform based on multi-DSP parallel algorithms.

Keyword: Software Radio; Digital Signal Processing (DSP); Parallel Algorithms

1. INTRODUCTION

As the software radio makes its transition from research to practice, it becomes increasingly important to establish provable properties of the software radio model on which product researcher can base technology insertion decisions. The main advantage of software radio is its flexibility and software-reconfigurable functionality, which can be used in different radio systems.

In the past, radio systems were normally designed and built to communicate using one or two waveforms. If two groups of people have different types of legacy radio they will not be able to communicate. This leads to problems in times of especial usage (such as in the case of war, e.g., if the Air Force was unable to communicate with the Army they are supporting, or during peace, e.g., in the police need to talk with other people assisting in a search and rescue). The need to communicate with people using different types of equipment can only be solved with software reprogrammable radios. One software radio can communicate with many different radios with only a change in software parameters.

In recent years, the concept of software-defined radio (so called SDR or SWR) has received much military, commercial, and academic interest. In its extreme form, a software radio would consist of digital hardware connected to the antenna. However, the implementation technology has not yet sufficiently matured for such applications, and near-future products will be based on less ambitious architectures.

One of the fundamental ideas of software radio is the expansion of digital signal processing toward the antenna, and thus to regions where analog signal processing has been dominant so far. It is straightforward to realize that the hardware platform is a most prominent enabling component of a software radio terminal. Of special interest is the very part of the terminal where analog signal processing is replaced by digital signal processing, namely the digital front-end.

A conventional approach to the implementation of a multi-standard radio is the utilization of multiple transceiver chains, each dedicated to an individual standard and optimized for that particular standard. On one hand, such an approach is not flexible, in the sense that most of the hardware needs to be replaced whenever the characteristics of the air interface change. Furthermore, this approach is becoming increasingly infeasible and economically unacceptable as the complexity of radio interfaces grows or the number of radio standards expands. On the other hand, a radio with software-reconfigurable functionality in every architectural layer could provide a more efficient approach toward the implementation of demanding multi-standard third-generation radios. For this reason, SWRs are considered one of the essential components of third-generation systems, and therefore have been experiencing ramped-up research and development over the last few years.

The key feature in SWR is the complete software-reconfigurability of the digital radio processor. This reconfigurability may be achieved either through software (different DSP subroutines) or electronically reconfigurable hardware (FPGA configuration files). The trade-offs between DSPs and FPGAs are fluid and technology-dependent. Many radio functions are best performed by DSPs while others are best performed by FPGAs.

This article explores the applicability of signal processing algorithms in software radio and digital signal processing in architecture of software radio. It also provides a basic implementation method for a multi-DSP array to fulfil some basic software radio functions.

2. DSP TECHNOLOGIES IN SOFTWARE RADIO

2.1 The requirements for digital processing in software radio

To a single channel, the software radio method which to use wideband A/D, DSP and the general-purpose CPU is less effective than to use hardware integrated techniques. The receiver which is integrated with hardware set is cheaper and

more suitable for ordinary application, such as home wireless, cellular mobile phone, etc.. However, software radio is provided with two advantages other than common radio system, that is, flexibility and complexity.

Among these features, the real implementation of SWRs has raised a number of technical problems to be solved, which have recently become a focal point in the technical literature. Most of these technical problems are related to analog-to-digital conversion (ADC) and digital signal processing (DSP) technologies, which in many cases cannot provide the advanced hardware needed to support the demanding SWR technical requirements. The potential of rapidly evolving these technologies and relaxing their limitations to enable the development of SWR has already been mooted throughout the world. However, no widely acceptable solutions have emerged so far.

2.2 Digital signal processing in an architecture of software radio

There is a difference if only one channel or multiple channels are to be selected off the input signal. In case of single-channel reception, the filters necessary for sample rate conversion (SRC) can be combined with the channelization filters. This eventually leads to highly efficient implementations that are vital for mobile terminal applications where power consumption is a major issue.

If multiple channels are to be received, the simplest approach is to use several one-channel channelization units in paralled. Still, by combining downconversion and channel filtering in filter banks the effort can be lowered, especially in narrowband systems. Particularly, in only signals of one standard are to be received, discrete Fourier transform (DFT) filter bands are promising candidates as channelizers, since all channels have the same bandwidth. Polyphase filter banks, one subclass of uniform DFT filter banks, have the desirable characteristic that the relative complexity tends to decrease as the number of channels to be separated increases, in contrast to the parallel implementation of one-channel-channelizers. Uniform DFT filter banks split the frequency band [0, fs] into an integer number of subbands. These subbands should represent the different channels. Thus, the sample rate fs must be an integer multiple of the number of channels. Still, earlier it was suggested that one choose the digitization rate standard independently fixed. Since, moreover, the channel spacing generally does not equal the symbol/chip rate, SRC is necessary before and after channelization. A solution to overcome the limitation of the integer ratio between the sample rate and the number of channels could be the application of nonuniform filter banks.

As mentioned above, a great challenge is the exploitation of commonalities of the digital processing and the signal processing algorithms required by the different standards of operation of the software radio terminal. In the current section we have seen that narrowband systems require narrowband channel filtering, and spread-spectrum systems, usually having a wide bandwidth, require dispreading. In this traditional narrowband receiver systems (*see Figure 1*), each channel is assigned with one RF receiver and IF receiver, in which RF receiver is used to transfer RF signal into IF signal, IF receiver to transfer IF signal into baseband signal. After being sampled in baseband, signal is conducted processing to produce voice signal.

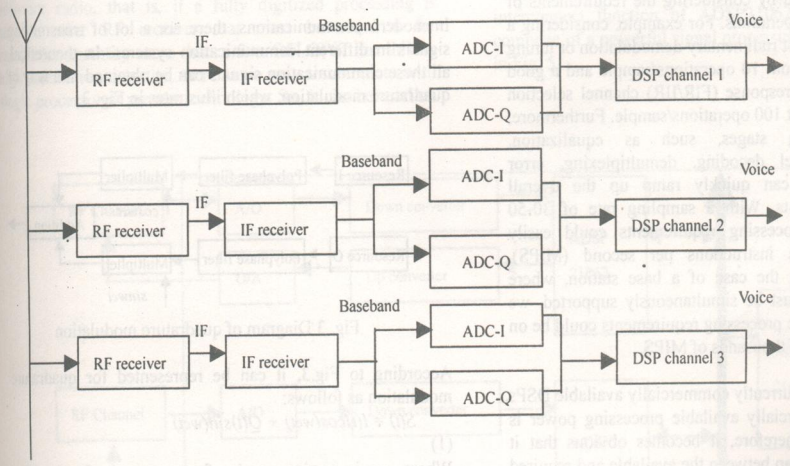


Fig. 1 Traditional narrow band receiver system (each with one front-end)

Fig. 2 illustrates the digital front-end of software radio terminals. In this way, all-channel shares one RF receiver, then IF signal being sampled using wideband ADC. The signal after sampled becomes involving all-channel digital IF signal, which is then getting into each independent of digital processing, complete channel selection, digital downconversion to baseband signal and conduct processing to produce voice signal. It is thus evident that all-channel share with digital front-end (RF and IF), cost decreases for each channel front-end.

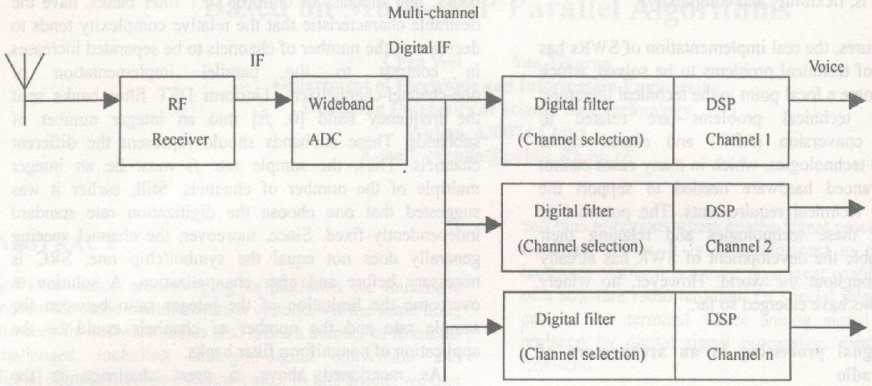


Fig. 2 Software implementation with sharing front-end

Following the ADC, the DSP domain in the generic SWR architecture illustrated in Fig.1 and Fig.2 should be capable of processing the digital data produced by the ADC and apply software means to implement a number of radio functions in real time. The main question arising in this context again concerns the structure of the DSP domain. In the simplest case, the DSP domain could be just a single DSP device; however, it is not likely that a single DSP could satisfy all the performance requirements, particularly for base station systems (take an example of cellular system). This is readily justified by considering the requirements of some typical system operations. For example, considering a single channel, the most rudimentary demodulation or tuning procedures requires about 10 operations/sample and a good finite/infinite impulse response (FIR/IIR) channel selection filter may require about 100 operations/sample. Furthermore, additional processing stages, such as equalization, deinterleaving, channel decoding, demultiplexing, error control, and so on can quickly ramp up the overall processing requirements. With a sampling rate of 30-50 MHz/channel, the processing requirements could easily surpass 5000 million instructions per second (MIPS). Moreover, considering the case of a base station, where dozens of channels must be simultaneously supported, we may infer that all of the processing requirements could be on the order of hundred of thousands of MIPS.

With reference to the currently commercially available DSPs, the maximum commercially available processing power is about 2000 MIPS. Therefore, it becomes obvious that it appears to be a large gap between the available and required processing resources. This lack of processing resources, known as the DSP bottleneck, raises another challenge in the development of SWR, which we refer to as the DSP challenge.

3. SIGNAL PROCESSING ALGORITHMS IN SOFTWARE RADIO

3.1 Modulation algorithms in software radio

Modulation signals in software radio are produced by different software method, which is based on general-purpose digital signal processing. Each modulation algorithm is made up of software model, while it is required to produce a certain modulating signal, only transfer relative model. Owing to the fact that different modulations are implemented with software, so in software radio, it may update the software of modulating model to match the ever-change of modulation system, which is of good flexibility and adaptability. All modulation algorithms in software radio can be implemented with DSP technology.

In modern communications, there are a lot of transmitting signals in different communication systems. In theoretical, all these communication signals can be obtained in a way of quadrature modulation, which illustrates in Fig. 3.

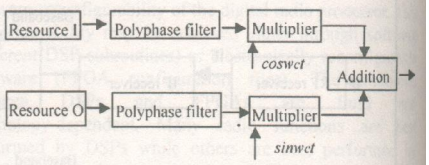


Fig. 3 Diagram of quadrature modulation

According to Fig.3, it can be represented for quadrature modulation as follows:

$$S(t) = I(t)\cos(wct) + Q(t)\sin(wct) \quad (1)$$

Where, wc is carrier angular frequency, $wc=2\pi fc$. After Equation (1) being digitized, it can be transformed as following equation (2) :

$$S(n) = I(n)\cos(nwc/ws) + Q(n)\sin(nwc/ws) \quad (2)$$

Where, ws is angular frequency of sampled frequency.

3.2 Demodulation algorithms in software radio

As above discussed, almost all functions in software radio are implemented with software, thus, demodulation is not exception. Demodulation of software radio is generally adopted with a method of digital coherent demodulating.

For continuous-wave modulation, a digital expression of modulated signal is as follows:

$$S(n) = A(n)\cos[w(n)n + \theta(n)] \quad (3)$$

Because there exists a definite relationship between frequency and phase, equation (3) can be transformed as following expression:

$$S(n) = A(n)\cos[wcn + \Phi(n)] \quad (4)$$

Where, wc is carrier angular frequency, so we can get the following equation (5):

$$S(n)=A(n)\cos[\Phi(n)]\cos(wcn)-A(n)\sin[\Phi(n)]\sin(wcn)$$
$$=XI(n)\cos(wcn) - XQ(n)\sin(wcn) \quad (5)$$

Where,
$$XI(n) = A(n)\cos[\Phi(n)]$$
$$XQ(n) = A(n)\sin[\Phi(n)] \quad (6)$$

This is what we want to get two quadrature parts, and on the basis of above $XI(n)$, $XQ(n)$, it may conduct demodulating for various modulation form.

4. SOFTWARE RADIO MODEL AND TRIAL-PLATFORM BASED ON MULTI-DSP PARALLEL ALGORITHMS

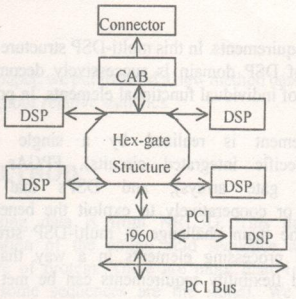
4.1 Basic concept of parallel processing in software radio

From above discussion we know, for implementing an ideal software radio, that is, if a fully digitized processing is adopted after RF receiver that must be required for calculation quantity more than a few GFLOPS. For this so high calculation quantity, there is no way to complete with single processor at present technology. So, it can be used on multi-DSP parallel algorithms to implement those functions.

4.2 Using DSP to realize Multi-processing parallel

Following is a typical processing method using multi-DSP to complete parallel algorithms applicable in software radio.

Figure 4 is Python/C6 structure diagram, which consists of 6 interfaces, four are C6x DSP dedicated , one is CAB and the other is i960 PCE interface.



CAB—Coreco Auxiliary Bus

Fig.4 the Hex-gate structure of Python/C6

4.3 Implementation of software radio model based on multi-DSP platform

In software radio, owing to the fact that various data must be dealt with and real time processing is required, generally a multi-DSP processing array is used to process front-end data in parallel. Following is an example of a multi-DSP method in software radio front-end processing. Due to the strongly interconnect function of ADSP21062, so it can simply comprise of a powerful signal processing array, as illustrated in Fig. 5.

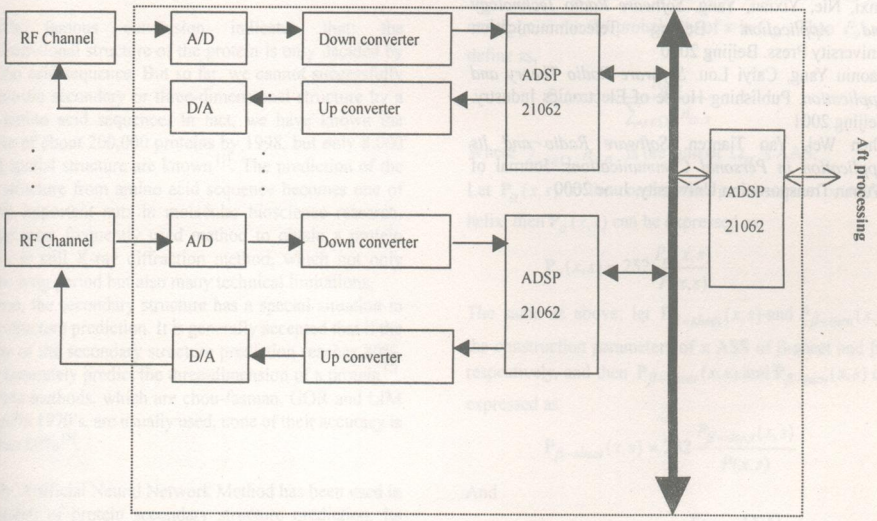


Fig. 5 Multi-DSP processing array based on ADSP21062

5. CONCLUSION

As a result of our investigations, we conclude that multi-DSP structure in SWR is an attractive platform for implementing signal processing where a single DSP could not satisfy all the

performance requirements. In this multi-DSP structure the functionality of DSP domain is successively decomposed into a number of individual functional elements. In practice, each
functional element is realized by a single ASICs (application-specific integrated circuits), FPGAs (field programmable gate arrays), and DSPs that work independently or cooperatively to exploit the benefits of parallelism. The main challenge in multi-DSP structures connecting all processing elements in a way that both processing and flexibility requirements can be met in an efficient way.

REFERENCES

[1] Joseph Mitola, *Software Radio Architecture: A Mathematical Perspective.* IEEE Journal on selected areas in communications. Vol. 17, No.4, April 1999

[2] Stephen P. Reichhart, *The Software Radio Development System.* IEEE Personal Communication. August 1999

[3] Ivan Seskar and Narayan B. Mandayam, *Softwware-Defined Radio Architectures for Interference Cancellation in DS-CDMA Systems.* IEEE Personal Communications. August 1999

[4] Apostolis K. Salkintzis, Hong Nie, and P. Takis Mathiopoulos. *ADC and DSP Challenges in the Development of Software Radio Base Stations.* IEEE Personal Communications. August 1999

[5] Xinxi, Nie, Yixian, Yang. *Software Radio Technology and Application.* Beijing Telecommunication University Press. Beijing 2000

[6] Xiaoniu Yang, Caiyi Lou. *Software Radio Theory and Application.* Publishing House of Electronics Industry. Beijing 2001

[7] Chen Wei, Yao Tianren. *Software Radio and Its Application in Personal Communications.* Journal of Wuhan Transportation University, June 2000

Supported by National Youth Teacher Funds [2000] 65

Secondary Structure Prediction Method Based on Neighbor Corrective Statistics

Chen Ming, Tong Genglei, Xu Jinlin and Luo Jianhua
Life Academy, Shanghai Jiao Tong University
Shanghai 200030, P.R.China
Email: l-jianhua@online.sh.cn

ABSTRACT

If the accuracy of the prediction of secondary structure is over 80%, we can predict the three-dimension of a protein. But, the accuracy of any prediction methods hasn't been over 70% by now. Now we put forward a new method based on the principle of neighbor relation statistics. Influence of neighbor amino acid can be taken into account to secondary structure. Suppose that the sample is big enough, our result is accurate. That is to say, our result is related with the amount of data. So up to the present, the result of the protein, which is mainly composed of helix, is better than the other only because our helix data is much bigger.

Keywords: Neighbor relation statistics, Amino Acid String, Structure Parameter, α helix, β-sheet and β-turn.

1. INTRODUCTION

As well known there are generally four kinds of protein structures:
a. The primary structure is the synonym of amino acid sequence
b. The secondary structure is periodical conformation mainly maintained by hydrogen bond. They are Alpha helix, Beta sheet and Beta turn. Some super-secondary structures are included, e.g. beta barrel.
c. d. third and fourth structure.

Anfisen's famous conclusion indicates that: the three-dimensional structure of the protein is only decided by the amino acid sequence. But so far, we cannot successfully point out the secondary or three-dimensional structure by a giving amino acid sequence. In fact, we have known the sequence of about 200,000 proteins by 1998, but only 8,000 of their spatial structure are known [1]. The prediction of the spatial structure from amino acid sequence becomes one of the most important nuts in molecular bioscience research. Now the most frequently used method to obtain a protein structure is still X-ray diffraction method, which not only takes too long period but also many technical limitations.
Of course, the secondary structure has a special situation in protein structure prediction. It is generally accepted that if the accuracy of the secondary structure prediction reaches 80%, we can accurately predict the three-dimension of a protein [2]. Now three methods, which are chou-fasman, GOR and LIM proposed in 1970's, are usually used, none of their accuracy is more than 60% [3].

Recently, Artificial Neural Network Method has been used in the research of protein secondary structure prediction. Its accuracy is 64% or so. Up to now, if homologous information of protein isn't taken into account, the precision of the best method is lower than 70%.

In this paper, we put forward a new method based on principle of Neighbor relative statistics.

2. METHOD

Our using data is from NRL3D. This database offers information of the amino acid sequence and secondary structure of protein. There are more than 14,000 proteins there (some sequences are the same). We eliminate all duplicated sequences and some illogical sequences in their structure information. We have counted the frequency of the amino acid emergence in secondary structure of 5000 proteins. In order to explain structure parameters, how to construct the α helix structure parameter based on neighbor relation statistics is taken as a example.

Let Ω be the statistical sample space, x be a kind of Amino Acid String (AAS) in Ω, n_x be the number of x in Ω, s be the length of AAS, and $P(x,s)$ be the probability of x in Ω. Then $P(x,s)$ is define as,

$$P(x,s)=\frac{n_x}{\sum_{x\in\Omega} n_x} \tag{1}$$

Where $\sum_{x\in\Omega} n_x$ is the sum of Ω 's AAS.

Let Ω_α be the Ω 's subspace which secondary structure are all α helix, $n_{\alpha,x}$ be the number of x kind of AAS in Ω_α, and $P_\alpha(x,s)$ be the probability of x in Ω_α. Then $P_\alpha(x,s)$ is define as,

$$P_\alpha(x,s)=\frac{n_{\alpha,x}}{\sum_{x\in\Omega_\alpha} n_{\alpha,x}} \tag{2}$$

Where $\sum_{x\in\Omega_\alpha} n_{x,\alpha}$ is the sum of Ω_α 's AAS.

Let $\mathbf{P}_\alpha(x,s)$ be the construction parameters of x ASS of α helix, then $\mathbf{P}_\alpha(x,s)$ can be expressed as

$$\mathbf{P}_\alpha(x,s)=252\frac{P_\alpha(x,s)}{P(x,s)} \tag{3}$$

The same as above: let $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-turn}(x,s)$ be the construction parameters of x ASS of β-sheet and β-turn, respectively, and then $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-turn}(x,s)$ can be expressed as

$$\mathbf{P}_{\beta-sheet}(x,s)=252\frac{P_{\beta-sheet}(x,s)}{P(x,s)} \tag{4}$$

And

$$\mathbf{P}_{\beta-turn}(x,s)=252\frac{P_{\beta-turn}(x,s)}{P(x,s)} \tag{5}$$

Respectively. Where the statistical sample space $\Omega_{\beta-sheet}$

and $\Omega_{\beta-turn}$ of $P_{\beta-sheet}(x,s)$ and $P_{\beta-turn}(x,s)$ are the subspaces in Ω which secondary structure belong to β-sheet and β-turn.

When s=2, there are 20^2 different permutation in twenty kinds of amino acids, or there are 400 data in all $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-turn}(x,s)$ respectively. When s=3, there are 8000 data in $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-turn}(x,s)$ sets respectively. The data is increased with the length s of AAS at 20 times.

Any amino acid sequence of protein can be changed into a data sequence of $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ or $\mathbf{P}_{\beta-turn}(x,s)$. For example, the amino acid sequence of protein

'NTFLSLRDVFGKDLIYTLYY'

can be converted to $\mathbf{P}_{\alpha}(x,s)$ sequence, or

46 46 79 95 64 155 136 78 90 33 70 16 85 127 43 46 49 79 72 59

So its spatial structure can be predicted according to $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-sheet}(x,s)$ sequence.

3. RESULTS & ANALYSIS

We have counted $P_{\alpha}(x,s)$, $P_{\beta s}(x,s)$ and $P_{\beta t}(x,s)$, where s=2,3, from amino acid sequences of 5000 proteins. We draw charts according to $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ or $\mathbf{P}_{\beta-turn}(x,s)$, as shown in Fig.1 (a). In Fig.1 (a), some peaks can be seen clearly and they are the certain secondary structures. The blue, purple and yellow curves in Fig.1 (a) are $\mathbf{P}_{\alpha}(x,s)$, $\mathbf{P}_{\beta-sheet}(x,s)$ and $\mathbf{P}_{\beta-turn}(x,s)$ respectively. In order to validate our method, we use the bio-software Anthepro (version 4.5) which provide some secondary structure predictions, that are the GOR I method (Garnier *et al.*, 1978), GOR II method (Gibrat *et al.*, 1987), Double Prediction Method (Deléage & Roux, 1987), "homologue method" (Levin *et al.*, 1986)

Fig. 1 is a structure prediction chart of a myoglobin protein which secondary structure is mainly composed of helix. And the color bar in Fig.1(b) to Fig.1(g) are used to be expressed right hand helix, beet-sheet and beta-turn secondary structure of proteins respectively.

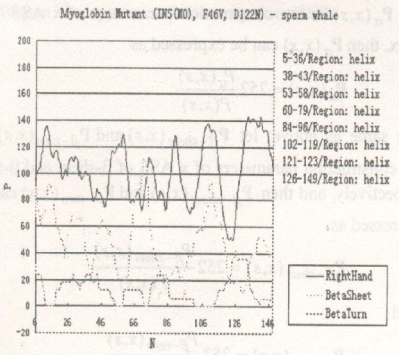


Fig 1 (a) The Result of Protein 101M

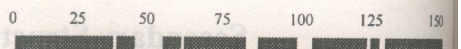


Fig 1(b) The Real Fact

Fig 1(c) Neighbor Corrective Statistics

Fig 1(d) GORI

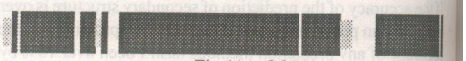


Fig 1(e) GORII

Fig 1(f) DPM

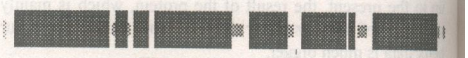


Fig 1(g) "homologue method"

The results of our method are compared with the other methods', as shown in Fig.1. The blue one in Fig.1 (b)-(g) means helix, the yellow one means beta sheet and the green one is beta turn. Using this way, their accuracy are **56.6%, 58%, 61.3%, 62.2%** respectively. The top figure means the amino acid location in sequence. The result of homologue method is best and our method is better than the other three. This method is based on neighbor relation statistics, so the more data we get, the more accurate the result is. Fig.2 (a) is the prediction curve of ferredoxin, which is almost no helix structure in (but some helix structure is provided in NRL3D). And main secondary structure is β-sheet in.

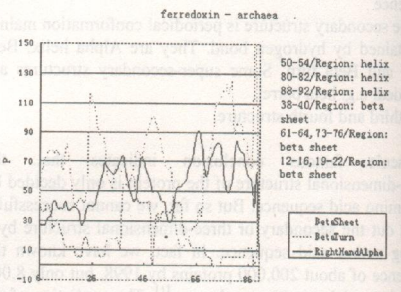


Fig 2(a) The Result of Protein 1xer

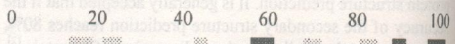


Fig 2(b) The Real Case

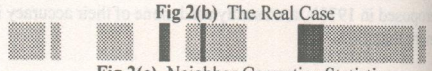


Fig 2(c) Neighbor Corrective Statistics

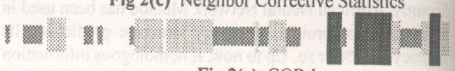


Fig 2(e) GOR I

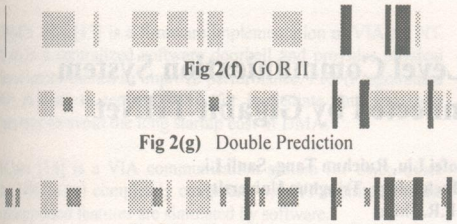


Fig 2(f) GOR II

Fig 2(g) Double Prediction

Fig 2(h) "homologue method"

The result is not as good as the first one shown in Fig.2. It is apparent that, any prediction results of all methods can't meet the fact shown in Fig.2 well. But the methods GOR I, II and DPM are not very terrible (terribly deviated), they can meet part of the real structure. But prediction results of our method and "homologue method" are the best two of all.

In Fig.3, the $P_\alpha(x,2)$ curve is so good that only 20-22 amino acid in the sequence can't be differentiated. But none of alpha 3-10 can't be pointed out in its 3-10 curve. The way is so simple that the mount of the data influences the accuracy of the results. It is only because that Ω_α is bigger than $\Omega_{\alpha 3-10}$. If we have big enough samples, or $\Omega_{\alpha 3-10}$ are big enough, the satisfying result will be got.

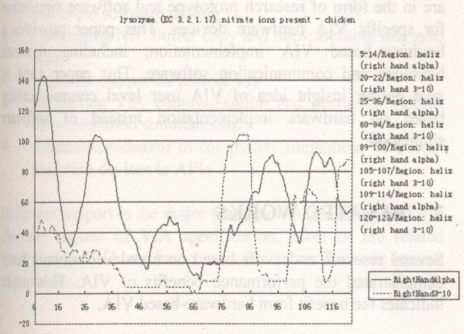


Fig.3 The Result of the Protein Lysozyme

In Fig.4, only helix structure is drawn. But this time our result of our method is not very satisfying. The reason is the helixes of this protein are too many, so we can't distinguish helix structure by the peak. This kind of protein can't be predicted by this method properly. The method is not suitable for this kind of protein, which is full of one type structure.

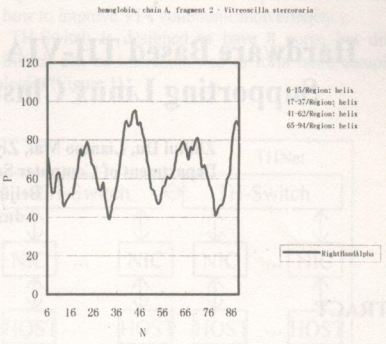


Fig 4 The Result of A Hemoglobin

4. CONCLUSION

We present new predict method of secondary structure on neighbor relation statistics. Influence of neighbor amino acid can be taken into account to the secondary structure. Almost all protein of which main secondary structure is helix has a better result than the other. But our result is not very satisfied sometimes. There are two reasons; first, we can't get a big enough sample space. Second, the information from database is subjective more or less because these data is obtained by X-ray diffraction method. That is to say, our result is related with the amount of data. So by now, the result of a protein, which is mainly composed of helix, is better than the other. The reason is our helix data is much bigger.

REFERENCES

[1]. Zhang C T, Zhang R. A New Criterion to Classify Globular Proteins Based on Their Secondary Structure Contents. Bioinformatics, 1998, 14(10): 319-344

[2].Rost B, Sander C. Prediction of Protein Secondary Structure at Better Than 70% Accuracy, J. Mol. Biol., 1993, 232:584-599

[3]. Wang ZX, the Present Situation and the Future of the Protein Structure Prediction, the Chemistry of Life, 1998, Vol. 18(6), 91-94

[4]. Sun Z. R, Rao X Q, Peng L W, Xu D. Prediction of Protein Supersecondary Structures Based on Artificial Neural Network Method, Protein Engineering. 1997,10(7):763-769

[5]. Rose G D, Wolfenden R. Hydrogen Bonding, Hydrophobicity, Packing, and Protein Folding. Ann Rev Biophys Biomeol Struct. , 1993, 22: 381-415

Hardware Based TH-VIA User Level Communication System Supporting Linux Cluster Connected by Gigabit THNet

Zhihui Du, Liantao Mai, Ziyu Zhu, Haofei Liu, Ruichun Tang, Sanli Li,
Department of Computer Science and Technology, Tsinghua University,
Beijing, 100084, P.R.China
duzh@tsinghua.edu.cn

ABSTRACT

VIA (Virtual Interface Architecture) is an open industry specification designed to facilitate the movement of distributed enterprise applications onto large-scale, high volume, distributed message passing clusters. VIA is seen as an interconnection technology to be used among clusters of PCs or workstations. This paper provides hardware supported VIA system that can provide high bandwidth and low latency communication on a custom gigabit crossbar network. This implementation is different from most other software or firmware VIA project in that our implementation is fully based on hardware. The design and implementation features are included in this paper. It is an academic endeavor in the emerging high speed SAN communication system.

Keywords: VIA, Gigabit Network, Cluster, User Level Network

1. INTRODUCTION

The capability of hardware networks [4-5] has grown from ten to hundred times, but the network software has not caught up with the explosive growth of hardware. ULN (User Level Network) protocols [3] try to improve the communication ability by removing the operating system from communication critical path and avoiding memory copy. Experiments of ULN protocols [6-10] show that the concept of ULN protocols is sound. Generally, ULN protocols are based on high speed and high reliable networks; these protocols are reduced compared with legacy protocols such as TCP/IP. VIA (Virtual Interface Architecture) [1-2] specification is an important initiative that aims to standardize the ULN protocols for SAN (System Area Network) to provide high bandwidth and low latency communication. VIA is jointly authored by Intel, Compaq and Microsoft and it has become an industry standard. Now more than 100 other industry leaders have joined and endorsed the collective endeavor.

VIA is a connection oriented communication model. The protocol is operating system and architecture independent. VIA gives user level processes direct but protected access to network interface cards. This allows applications to bypass IP processing overheads (copying data, computing checksums, etc.) and system call overheads while still preventing one process from accidentally or maliciously tampering with or reading data being used by another.

VIA is comprised of four basic components: Virtual Interface, Completion Queues, VI Providers, and VI Consumers. The VI Provider is composed of a physical network adapter and a software Kernel Agent. The VI Consumers are composed of applications and operating system communication facility. The Virtual Interface is the mechanism that allows a VI Consumer to directly access a VI Provider to perform data transfer operations. A Completion Queue allows a VI Consumer to coalesce notification of Descriptor completions from the Work Queues of multiple VIs in a single location.

To our knowledge, there is no commercial VI implementations exists to now. All implementations of VI are in the form of research prototype and software simulation for specific VIA hardware devices. This paper provides a hardware-based VIA implementation, including network hardware and communication software. This paper aims to give deeper insight idea of VIA user level communication based on hardware implementation instead of software simulation.

2. RELATED WORKS

Several research endeavors based on user-level network have demonstrated the performance benefits of VIA. This paper indicates the benefit from hardware-based VIA.

Berkeley VIA [11] is a software simulative implementation of VIA, and it changes the APIs of VIA in some degree. It is a partial implementation and it uses NIC memory to implement a pair of doorbells (software doorbell). Memory registration is implemented by allocating fixed amount of kernel space memory and mapping it to the user space. Different extensions (active doorbells, fast descriptors, broadcast VI) on VIA have been discussed.

M-VIA (Modular-VIA)[12] is a complete software implementation of the VIA for Linux. M-VIA does not require hardware support for VIA, although it will take advantage of such support. M-VIA consists of a loadable kernel module and a user level library, and also requires a modified device driver (which is also a loadable kernel module). M-VIA coexists with traditional networking protocols. Without hardware support, M-VIA is zero-copy (no memory-to-memory copy) on sends but it needs to trap into kernel space (fast trap), and one memory copy on receives.

IBM's VIA [13] is a firmware implementation of VIA for NT. It uses a centralized software doorbell and provides physical descriptors caches to improve performance. PIO (Programmed IO) is used to replace DMA for immediate data and short message to avoid the long startup cost of DMA.

SCnet [14] is a VIA communication system for Unix. Since the NIC is not completely compliant with VI Architecture, the unsupported features are simulated by software.

[15] discusses the scalability in VIA implementation. It uses a detailed simulation model to show the different hardware and software alternatives in implementing VIA, at the same time, provides the methods of improving the scalability of VIA by firmware and hardware enhancements.

All the works are important contributions to VIA research. But none of them are based on hardware implementation. This paper gives a hardware-based implementation of VIA communication system.

3. CONTRIBUTIONS

The contributions of this paper include the following aspects.

- Supporting hardware doorbell to improve performance and avoid software polling
- Eliminating NIC buffer copy and implementing true zero copy
- Hardware supported protect VIs
- Supporting VIA communication and pervasive TCP/IP communication simultaneously
- Academic endeavor to completely implement VIA from hardware devices to APIs

Hardware support is the major feature of this research and it is also the request of VIA specification. Most of the related researches on VIA are based on software or firmware instead of hardware.

4. THNet

THNet is a Gigabit crossbar switch network which provides hardware support of VIA. The switch (we name it as TH-Switch) of THNet contains a 300,000-gates FPGA and each NIC (Network Interface Card, we named it as TH-NIC) a 30,000-gates FPGA. LVDS is used in THNet links. Currently each switch has 8 ports and the TH-NIC support PCI specification 2.2 that supports 66MHz and 32bits data transmission (2.112Gps).

NIC device that can support VIA specification is so complicated and expensive that most of the VIA researches use Myrinet [2,16-17] to simulate a VIA device. The advantage of this method is that it can give some useful results quickly; the disadvantage is that it cannot give us deeper insight of the hardware VIA implementation.

THNet aims to support VIA in hardware level, and gives directly research results on how to implement hardware VIA and how to improve VIA communication efficiency.
The TH-Switch is designed to have 8 ports, but different TH-Switch can be interconnected to form more complicated topologies (Figure 1).

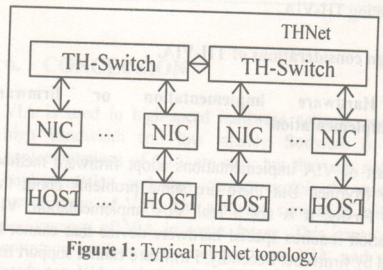
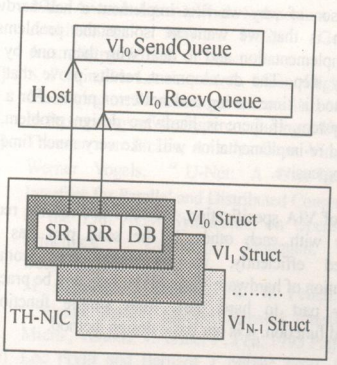


Figure 1: Typical THNet topology

The TH-NIC implements hardware doorbell, hardware supported send queue and receive queue. Different hardware devices support the connections between different VIs and each VI is protected when it is created and provided to user process. For now, each TH-NIC supports 8 VIs (Figure 2, N=8). The next version of the TH-NIC will support more hardware VI devices (64 hardware VI devices or more). The first goal of our project is to design and implement a small and practical hardware VIA network, and then extend its scale. In the TH-VIA implementation, VI_0 is used for TCP/IP communication. The IP package from upper layer is transmitted as a message of VI_0. Both VIA and TCP/IP communication are completely supported by THNet.

The process directly sends message from user process space (which has been mapped with kernel and TH-NIC memory) to network, so no extra memory copy is needed (true zero copy). To receive data, a FIFO is needed to synchronize the clock. The received data flow through the FIFO and then is copied into the host memory directly without any store and forward operation.



SR: Send Register
RR: Recv Register
DB: Doorbell

Figure 2: Each NIC has N VI hardware devices

5. TH-VIA DESIGNS & IMPLEMENTATION

TH-VIA completely supports the VIA specification and all the requested VIA APIs are implemented. In this paper, we mainly provide the hardware related features on designing and implementing TH-VIA.

5.1 Design considerations of TH-VIA.

5.1.1 Hardware implementation or firmware implementation

Now most of VIA implementations adopt firmware method, typically Myrinet. But there are some problems about this method. Firstly, it is not a real VIA implementation. VIA specification requires special hardware device that cannot be provided by firmware. Secondly, Firmware cannot support true zero copy (Figure 4). The message has to be copied from host memory into NIC buffer, and from NIC buffer to host memory. Memory copies will cause performance lost. Hardware implementation can avoid this kind of copy. Thirdly, FPGA is simple, powerful, flexible and easy to implement. Embedding a processor into NIC will need more development time and complicated development environment.

5.1.2 Complete hardware implementation or half hardware implementation at first

Based on the VIA specification, much more hardware function is needed to satisfy the function partition of hardware and software. The time of hardware development is much longer than the time of software. To implement a complete VIA system quickly and to overlap the software and hardware development time as long as possible, we decided to implement a half-hardware VIA system at first. The system will provide the hardware function on the send side. On receive side only part of the hardware functions are supported and the rest functions are simulated by software. The total hardware support of receive phase is put to next step.

Another reason of why we first implement a half-hardware VIA system is that we want to isolate the problems in hardware implementation and to deal with them one by one and step by step. The development results prove that this design method is time saving and not error prone. For a new hardware system, if there is hardware design problem, the redesign and re-implementation will take very much time and the cost is expensive.

In the eye of VIA specification, the send and receive request are similar with each other. If the send part has been implemented efficiently, in the next step, complete implementation of hardware VIA will be easy and be practical. From easy part to hard part, from simple function to complicated function, it is our basic design method.

5.1.3 Design of descriptors

The design aims of TH-VIA descriptors include two aspects. The first is compliant with VIA specification and the second is easy to be implemented by hardware.

TH-VIA divides the structure of a descriptor into two parts. One part is the structure defined by VIA specification, the other part is the condensed and efficient structure provided by hardware implementation. TH-VIA library analyses the descriptors provided by user applications and transforms them to the format needed by the TH-NIC. Two parts design combines portability with efficiency together.

5.2 Implementation

5.2.1 Hardware Doorbell

In this implementation, hardware doorbell is a counter. When a user process pushes its doorbell, the NIC automatically add the counter with 1.

The NIC polling the different counters of different user processes, when it finds any counter not equal to 0, it will get the corresponding descriptor and execute the corresponding message transmission. The NIC reduces the value of doorbell counter with 1 when it has dealt with the descriptor. Figure 3 gives the working mechanism of doorbell.

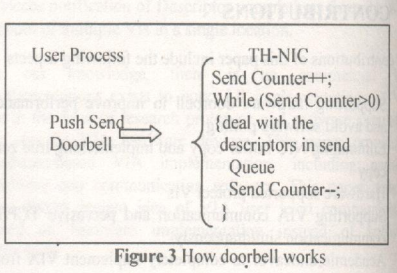


Figure 3 How doorbell works

5.2.2 TH-NIC memory mapping

The TH-NIC has 8 queues (corresponding to 8 VIs). The size of each queue is equal to 64 items and the size of each item is 32 bytes. Table 1 gives the memory mapping relationship between host memory and TH-NIC memory.

The memory of TH-NIC is mapped into host memory so that the user process can access them directly. In table 1, the send and receive pointer give the host memory position which serve as message buffers. The addresses from Base+8 to Base+15 are eight doorbell counters. Pushing the doorbell makes the counter add its value with 1. The TH-NIC polls these counters when it is idle.

Table 1: TH-NIC memory mapping

Address	Unit	Mode	Function
Base+0	2 words	Write (in 32 bits)	Receive pointer.
Base+4	2 words	Write (in 32 bits)	Send pointer.
		Read	Clear after read
Base +8	Byte	Write	counter+1

Address	Unit	Mode	Function
		Read	counter
...			
Base+15	Byte	Write	counter+1
		Read	counter

5.2.3 True Zero Copy

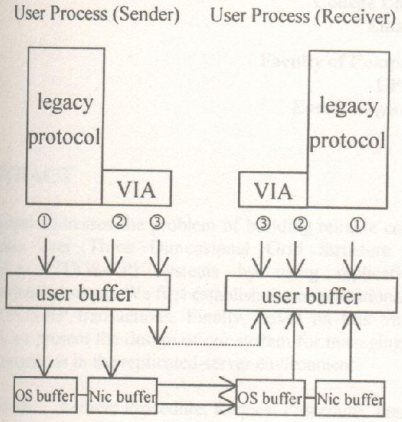


Figure 4 Comparison of legacy protocol, firmware VIA and hardware VIA

① Legacy protocol ② Firmware VIA

③ Hardware VIA

TH-VIA implements the true zero copy. Figure 5 shows the differences in different kinds of communications. The legacy protocol (TCP/IP) has 4 memory copies and complicated protocols, firmware VIA (Myrinet based VIA) has 2 memory copies and TH-VIA has not any memory copy. Memory copy is one of the major software overheads so true zero copy can improve the communication significantly.

5.2.4 Hardware Latency

The hardware latency of THNet is shown as in table 2. Because the total system has not been finished, we only provide the hardware latency data here. The total communication latency and bandwidth will be tested in the next step. For immediate data, the hardware latency is about 40/66=0.6μs, for message whose length is Len, the hardware latency is about (47+Len)/66μs. which is much faster than firmware implementation.

Table 2: Hardware Latency (Unit: CLK)

Phases	Immediate Data	Buffer
Doorbell	2	3
Send descriptor	3	5
Link setup	15	15
Package header	5	5
Transmission	10	14+Len
Receive	3	6
CRC	2	2
Total	40	47+Len

6. CONCLUSION

VIA is used in high-speed hardware network and provides high bandwidth and low latency. Software or Firmware implementations of VIA are easy but they are not real VIA implementation and they cannot exploit the complete the performance of VIA. The software simulation will lose the performance of VIA in some degree. This paper presents a prototype implementation of hardware VIA, and shows that hardware VIA is better than firmware VIA say nothing of software VIA. Hardware VIA is feasible and it can provide the complete benefit of VIA and hardware VIA is the direction of future VIA implementation.

REFERENCES

[1] Virtual Interface Architecture Specification. http://www.viarch.org/.

[2] D. Dunning et al., "The Virtual Interface Architecture," *IEEE Micro*, Mar.-Apr. 1998, pp. 66-76.

[3] Bhoedjang, R.A.F., et al., "User-Level Network Interface Protocols", IEEE Computer, November 1998, pp. 53-60.

[4] Nanette J.Boden, Danny Cohen, Robert E.Felderman, Alan E.Kulawik, Charles .L.Seitz,Jakov N.Seizovic, Wen-King Su, " Myrinet: A Gigabit-per-Second Local Area Network" , IEEE Micro, Feb. 1995: 29-36.

[5] Giganet. http://www.giganet.com.

[6] T. von Eicken, D. E. Culler, S. C. Goldstein, and K. E. Schauser, "Active Messages: a Mechanism for Integrated Communication and Computation" , Proceedings of the 19th Annual International Symposium on Computer Architecture, May 1992, Gold Coast, Australia.pp256-266

[7] S. P akin, M. Lauria, and A. Chien. High Performance Messaging on Workstations: Illinois Fast Messages (FM). In Proceedings of the Supercomputing, 1995.

[8] Thorsten von Eicken, Anindya Basu, Vineet Buch, and Werner Vogels, " U-Net: A User-Level Network Interface for Parallel and Distributed Computing ",Proc. of the 15th ACM Symposium on Operation Systems Principles, Copper Mountain, Colorado, December3-6,1995:40-53

[9] M. Blumrich, C.Dubnichi, E. W. Felten, and K. Li. "Virtual Memory Mapped Network Interfaces" . IEEE Micro , Volume: 15 Issue: 1 , Feb. 1995 Page(s): 21 -28

[10] Loc Prylli and Bernard T ouranc heau. BIP: A New Protocol Designed for High Performance Networking on Myrinet. In Proceedings of the International Parallel Processing Symposium Workshop on Personal Computer Based Networks of Workstations, 1998.

http://lhpca.univ-lyon1.fr/.

[11] P . Buonadonnaa, A. Geweke, and D.E. Culler. An Implementation and Analysis of the Virtual Interface Architecture. In Proceedings of the Supercomputing (SC) , pages 7-13, Nov. 1998.

[12] M-VIA: A High Performance Modular VIA for Linux. http://www.nersc.gov/research/FTG/via/.

[13] M. Bazikazemi, V. Moorthy, L. Herger, D. K. Panda, and B. Abali. Efficient Virtual Interface Architecture Support for IBM SP Switch-Connected NT Clusters. In *Proceedings of International Parallel and Distributed Processing Symposiu*m, May 2000.

[14] Ogawa, N.; Kurosawa, T.; Tachino, N.; Savva, A.; Fukui, K.; Kishimoto, M. "Smart Cluster Network (SCnet): design of high performance communication system for SAN". Cluster Computing, 1999. Proceedings. 1st IEEE Computer Society International Workshop on , 1999 Page(s): 71 -80

[15] Nagar, S.; Sivasubramaniam, A.; Rodriguez, J.; Yousif, M. "Issues in designing and implementing a scalable virtual interface architecture". Parallel Processing, 2000. Proceedings. 2000 International Conference on , 2000 Page(s): 405 -412

[16] Bhoedjang, R.A.F., et al., "User-Level Network Interface Protocols", IEEE Computer, November 1998, pp. 53-60.

[17] von Eicken, T.; Vogels, W. "Evolution of the virtual interface architecture", Computer , Volume: 31 Issue: 11 , Nov. 1998 Page(s): 61 –68

Managing Replicated Remote Procedure Using Three Dimensional Grid Structure Protocol

M.Mat Deris, A. Mamat*, M.Y. Saman*, H. Ibrahim*
Department of Computer Science
Faculty of Science and Technology
College University Terengganu, Malaysia
Email: mustafa@uct.edu.my

Faculty of Computer Science and Information System
UPM, Serdang, Malaysia*
Email: ali,yazid,hamidah@fsktm.upm.edu.my

ABSTRACT

This paper addresses the problem of building reliable computing programs over Three Dimensional Grid Structure Remote Procedure (TDGS-RP) systems by using replication and transaction technique. We first establish the computational model: The TDGS-RP transactions. Finally, based on this transaction model, we present the design of our system for managing TDGS-RP transactions in the replicated-server environment.

Keywords: Remote Procedure, Request Procedure, Transaction, Replication, Distributed Database.

1. INTRODUCTION

A Remote Procedure Call (RPC) model is the most popular model used in today's distributed software development and has become a *de facto* standard for distributed system in supporting fault tolerant computing [15]. The RPC proposed by [15] is a combination of remote procedure, replication, and transaction management to form a reliable, available and efficient distributed computing environment.

Transaction Management is a well-established concept in database system research. A transaction can be defined as a sequence of operations over an object system, and all operations must be performed in such a way that either all of them execute or none of them do [7]. Transactions are used to provide reliable computing systems and a mechanism that simplifies the understanding and reasoning about programs.

Replication is a useful technique to provide high availability, fault tolerance, and enhanced performance for distributed database systems [4,5,9] where an object will be accessed (i.e., read and written) from multiple locations such as from a local area network environment or geographically distributed worldwide. For example, student's results will be read and updated by lecturers of various departments. Financial instruments' prices will be read and updated from all over the world [14].

However, the replication protocol being used in RPC is analogous to Read-One Write All (ROWA) protocol: if one site is not accessible, the processing of an object is noted in the *partial commit* state, and resolved it after some time delay. This will increase the response time (one of the major performance parameter [11]), and therefore decreases the performance of the system. For the case of availability, RPC provides restricted availability of update operations since they cannot be executed (normal state i.e., commit or abort) at the failure of any copy. Nevertheless, the TDGS protocol proposed by [10] shows better performance when compared with ROWA since they can still execute if *alive* replicas are sufficient to form a Three Dimensional Grid Structure (TDGS) quorum where update operations have high availability with minimum communication cost. In the Appendix, we show how the communication cost and availability of the TDGS and ROWA are being carried out, followed by a performance comparison between them.

In this paper, a model analogous to the RPC proposed by [15] is presented. However a transaction processing algorithm of RPC will be changed by adopting the TDGS protocol when the number of replicas, n≥8. Otherwise, voting protocol will be adopted using restructuring logical structure proposed by [1]. This paper will discuss only cases where the number of replicas, n≥8. The proposed model is called TDGS remote procedure (TDGS-RP) model.

The remainder of the paper is organized as follows. Section 2 introduces the replicas, the TDGS-RP model, and the transaction model. Section 3 presents the model and the algorithm for transaction management. Section 4 described the replica management. Section 5 and 6 present the correctness and examples of the proposed system, respectively. Section 6 concludes the paper.

2. SYSTEM MODELS

2.1 Replicas

A distributed system consists of sites or servers interconnected by a communication network. A service is provided by a group of servers (called *replicas*) executing on some sites. These replicas manage some common data objects that can be shared by users/clients. We assume that each replica knows the location of other replicas that store the same data objects.

Let $C = \{C_1, C_2 \ldots C_n\}$, where $I = 1,2 \ldots n$ be a set of replicas.

Each replica, C_i manage a set of data objects: $D_i = \{d_1^i, d_2^i, \ldots, d_m^i\}$. The consistency constraint requires that $D_i \equiv D_j$, for i,j = 1,2,…,n.

2.2 The TDGS-RP Model

Each replica in the system provides a number of request procedures that can be called by clients for processing the data objects managed by the replica [15]. We use R to denote the set of all request procedures provided by all replicas in the system:

$$R = \{ r \mid r \text{ is a request procedure of the system} \}$$

After we make a request procedure, the request may return successfully or failed. There are reasons for the failure of a request, such as a server error, communication error, or the object is not free. We define the effects of an TDGS- RP as the processing of an object. Hence we can abstract a TDGS-RP as a mapping below:

$$r: P \times D \rightarrow \{SUC, FAIL, PC\}$$

Where D is the union of all data objects. The values of the target set have the following meaning:

SUC This means that no failure occurred during the TDGS-RP's execution. By $r(p,d) = SUC$, where $p \in P$ is a procedure and $d \in D$ is the data object processed by the TDGS-RP, we mean that the TDGS-RP was successful (the procedure performed the job).

FAIL This means accessing failure. By $r(p,d) = FAIL$, we mean that the server could not perform the job because, for example, the object managed by the server is not free. This means that the TDGS-RP has not been executed.

PC This means partial commit state. By $r(p,d) = PC$, we mean that the network is partitioned into sub-networks.

2.3 The TDGS-RP Transaction Model:

We define a TDGS-RP transaction as $T = \{r(p,d)\}$, where $r(p,d)$ is an TDGS- RP, and $p \in P$, and $d \in D$. After issuing the transaction, r of T will be executed if no error occurs (commit or SUC). If it fails, it will be rolled back (abort or FAIL): The TDGS-RP $r(p,d) \in T$ returns a SUC if and only if there exist a TDGS quorum of d that has successfully performed the procedure p. Conversely, $r(p,d)$ returns FAIL if no TDGS quorum of d has successfully performed the procedure p.

In addition to the two normal states, we define a *partial commit* (PC) state when a network is partitioned. During the conflict resolution phase, all transactions that returned a PC will be checked. If that transaction does not conflict with any other transactions (it does not use any common data objects with other transactions), then it will commits. If a conflict is detected between two transactions, one of them is then chosen as a victim and will be aborted. The other will then be safe and committed.

In the case where a replica is not accessible (e.g. replica is down), the transaction has only performed TDGS-RPs on other replicas that are alive. However, when the down replica recovers and re-join the service, these transactions will be made to commit and will update all its stale information.

3. TDGS-RP TRANSACTION MANAGEMENT

3.1 The TDGS-RP Transaction Processing Model

We define the *primary replica* for a data object d as a replica that is the nearest site in performing a TDGS-RP $r(p,d)$. Three system components are involved in processing a transaction submitted by a user/client.

- A TDGS-RP transaction manager (RTM) accepts a transaction T from the client. The RTM sends TDGS-RP $r(p,d) \in T$ to a primary replica of d and asks the primary replica to check if the TDGS-RP replica can be performed or not. The algorithm will be described later.
- A primary replica accepts, from the RTM, an TDGS-RP $r(p,d)$ and checks the executability of the TDGS-RP (the $a(r(p,d))$ operation). The primary replica sends the TDGS-RP to all replicas of data object d. We use $b(r(p,d))$ to represent this operation. The replica then cooperates with the primary replica by returning the executability check. The algorithm will be described later.
- Each replica of TDGS quorum of the data object d accepts, from the primary replica, the TDGS-RP $r(p,d)$ and the request to check the executability of the TDGS-RP (the $b(r(p,d))$ operation). The replica then cooperates with the primary by returning the executability check.

Checking the executability of a TDGS-RP is essentially a request for a lock on the data object. The actual effect of an $a(r(p,d))$ depends on the associated $b(r(p,d))$. That is, if the TDGS quorum of $b(r(p,d))$ can be formed (returns a SUC) then $a(r(p,d))$ returns a SUC. Otherwise $a(r(p,d))$ returns a FAIL. Figure 1, depicts the model of TDGS-RP transaction processing. If there are PCs returned by $b(r(p,d))$ calls, then $a(r(p,d))$ returns PC.

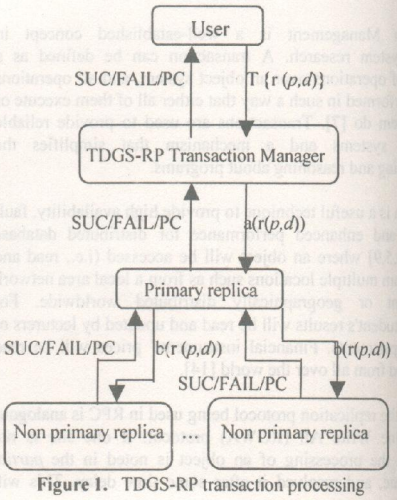


Figure 1. TDGS-RP transaction processing

A client submits a transaction to the RTM and waits for the resulting state, i.e., SUC, FAIL, or PC. The next operation will continue as soon as the client receives a resulting state SUC or FAIL. However, if the transaction returns a PC state, the conflict resolution will be resolved due to the network partitioning.

3.2 The Transaction Manager

The transaction manager has two basic functions to perform when it receives a transaction T:

- It submits operation to their primary replica. The manager is responsible for locating primary replica for the TDGS-RP and then sending these TDGS-RP to the primary replica.
- It manages the atomicity of the transaction T using the two-phase-commit (2PC) protocol to ensure that if the TDGS-RP of T to the primary replica returns a SUC, it will commit and returns a SUC. If the TDGS-RP returns a FAIL, it asks the primary replica to abort and returns a FAIL. In this case, any uncommitted transaction will eventually be rolled back. If there are some operations return a PC, then T asks primary replica to partially commits and returns a PC.

Algorithm 1- The RTM Algorithm

```
Manage_TDGS-RP_Transaction_manager()
{
 Boolean Incomplete = TRUE;
 while (TRUE)
 {
 //receive TDGS-RP transaction
 // If more than one transactions come, queue up
 //them until the processing of the current
 // transaction is over
 receive (client, T);
 primary_replica = server for client that invoke T;
 ret = a(r(p,d));
 execute on primary_replica;
 }
    S_SUC ={a(r(p,d))| ret= SUC};
    S_FAIL ={a(r(p,d))| ret = FAIL};
    S_PC   ={a(r(p,d))| ret = PC};
    Switch
    {
     case (S_SUC ≠ 0)
     // a(r(p,d)) executed and returns SUC
      send (primary_replica, "commit");
     tell the client that transaction returns SUC;
      Incomplete = FALSE;
      break;
      case (S_FAIL ≠0);
      // a(r(p,d)) returns FAIL
       send (primary_replica, "abort");
       tell the client that transaction returns FAIL;
      Incomplete = FALSE;
       break;
      case (S_PC ≠0);
      // no failure, but there are PC returned
      set asynchronous receive procedure handler;
                each primary replica is given the address
                of the handler;
               // address handler  is used during the
               // recovery process
               send (primary_Address[i], "partial commit");
               // i  is the number of partitions
               tell the client that transaction returns PC;
               Incomplete = FALSE;
               break;
                    }
                }
```

4. REPLICA MANAGEMENT

4.1 The coordinating algorithm for the primary replica

When a primary replica receives an TDGS-RP r(*p,d*) from a transaction manager, it uses the coordinating algorithm to maintain the consistency of all replicas in terms of TDGS-RP. This section describes the coordinating algorithm.

In the coordinating algorithm, the primary replica uses the 2PC protocol to ensure the replication consistency. In the first phase, the primary replica asks the TDGS quorum whether it can be formed or not (for b(*r*(*p,d*)) operation). If the TDGS quorum can be formed (replicas under TDGS quorum return a SUC for such execution), the primary replica returns a SUC to the transaction manager. If the transaction manager requests a commit, then in the second phase the primary replica asks all replicas to commit the TDGS-RP execution. If TDGS quorum cannot be formed, then the primary replica returns a FAIL to the transaction manager and asks all replicas to abort the operation in the second phase. If operation returns a PC (the primary replica returns a PC and other non-primary replicas may return a PC), the primary replica returns a PC to the transaction manager. The primary replica also records the number of replicas that return a PC. This number will be used in conflict resolution during the recovery process. The coordinating algorithm is listed in Algorithm 2. We assume that C_m is the primary replica.

Algorithm 2- The Coordinating Algorithm

```
Primary_replica_process()
{
 Boolean Incomplete = TRUE, Network_Status=TRUE,
                      StatusPlane = FALSE;
// network_Status = FALSE if network is partition into
// several parts
 int i,j;
 while (TRUE)
 { int i,j;
   // where j is the number of hypotenuse replicas
   //receive TDGS-RP from RTMs
   // If more TDGS-RPs come, queue up them
   get the status of the network;
   if network is partitioning
      get  the number of replicas, say k;
       Network_Status = FALSE;
    endif
    receive (RTM, a(r(p,d)));
    Let C_m , m< k be the primary_replica;
```

```
While (Incomplete)
for i=1 to 4;
for j=1 to 2 ;  // hypotenuse replicas

   HR(i,j) = b(rij(pij,dij));
   // replica j of hypotenuse replica i.
   Hypotenuse_Replica(i)= HR(i,1) ∧HR(i,2);
   // return SUC if HR(i,j), j =1 ,2 returns SUC,
   // else FAIL
   if Hypotenuse_Replica(i) ≡ SUC
     Incomplete = FALSE;
endwhile
HR_SUC = ∃Hypotenuse_Replica(i) ≡ SUC ∧
              Network_Status = TRUE, i=1,2,3,4;
HR_FAIL = ∀Hypotenuse_Replica(i) ≡ FAIL,
              i=1,2,3,4;
HR_PC = ∃Hypotenuse_Replica(i) ≡ SUC ∧
            Network_Status = FALSE, i=1,2,3,4;
Switch {
 Case (HR_SUC≠0);
  Do while (StatusPlane = FALSE) {
    for i=1 to 4
    for j=1 to 4
    // each plane consists of 4 edges (replicas)
     edges(i,j) = b(rij(pij,dij));
     // replicas that are edges to plane i.
     Plane(i) = Πedges(i,j), i=1,2,3,4;
     // returns SUC if  edges(i,j), j=1,2,3,4
      returns SUC, else FAIL
    If Plane(i) ≡ SUC
    tell the RTM that the first phase returns SUC;
    StatusPlane = TRUE;
   else
    tell the RTM that the first phase returns FAIL;
  }
 enddo
 case (HR_FAIL≠0)
  Tell the RTM that the first phase returns FAIL;
      break;
    case (HR_PC ≠0)
      Tell the RTM that the first phase returns PC;
      NoOfPCs = k;
      break;
    }
  receive(RTM, command);
  switch{
    case ( command is, "commit")
      send(Si, "commit"), i=1,2,…,k;
      break;
    case ( command is, "partial commit")
      send(Si, "partial commit", NoOfPCs),
           i=1,2,…,k;
      break;
    case (command is, "abort")
      send(Si, "abort"), i=1,2,…,k;
      break;
  }
 }
}
```

4.2 The Cooperating Algorithm for TDGS Replicas

The TDGS replica is a set of replicas that consists of one of hypotenuse replicas[1] and edges (replicas) from each plane. When a TDGS replica receives a request from a primary replica, it check whether the request can proceed or not and acts accordingly.

- If the request can be performed, the TDGS replica locks the required data object and returns SUC. Later if the primary replica asks to commit the operation, the hypotenuse replica performs the operation and releases the lock. If the hypotenuse replica asks to abort the operation, it will release the lock.
- If the hypotenuse replica finds that the operation cannot be executed (the required data object is not free), it then returns an FAIL.
- If the TDGS replica finds that the data object is in a partially committed state, then it returns a PC to the primary replica. The primary replica will then partially commits the operation and record the event when the primary replica asks to partially commit the operation. If the primary replica asks to abort the operation, then the non-primary replica aborts the operation.

The 2PC protocol used by TDGS-RP transaction manager guarantees that the replicas will be in a consistent state if a transaction returns a SUC or FAIL. However, to guarantee all replicas do the same thing to the transactions with PC pending, a majority consensus should be reached among all replicas. We use a Resolution (RES) table, which essentially is a checkpointing log to record the events of partially committed RPCs. The RES table structure is listed in Listing 3.

Listing 3: The RES Table

```
Typedef struct res {
Char *rpc; // name of the partially committed RPC
Char * data; // data object name
int pc;
 // number of replicas partially committed this RPC
  void *b_image;
  // before image,of the data object
  void *handler;
 // asynchronous receive handler address
  } RES;
```

When the primary replica asks for a partial commit for an RPC, all replicas (including the primary replica) will record this event into its own RES table as a new entry. If t∈RES is such an entry, then *t.rpc* contains the name of the RPC, *t.data* contains the data object used in the RPC, and *t.pc* stores the number of replicas that have partially committed this RPC. This number (*t.pc*) is sent by the primary replica to each replica when it asks for partial commit (see Algorithm 2). It will be used during the conflict resolution algorithm to determine which a partial commit should be upgraded to a commit or which partial commit should be downgraded to an abort if a conflict occurs. In order to

[1] A pair of replicas from the hypotenuse edge in a box-shape structure of replicas is called hypotenuse replicas. Details can be found in [10].

downgrade a PC to an abort, a *before image* of the data object is kept in the RES table. *t.b_image* is used to record the address of the before image.

We associate with each data object d a lock called *d.lock*. The actual effect of b(r(*p,d*)) on *d* is to check or change the values of *d.lock*. That is, if *d.lock* = LOCKED(-1), b(r(*p,d*)) returns FAIL. If *d.lock* = FREE, b(r(*p,d*)) returns a SUC and set *d.lock* = LOCKED.

All TDGS-RPs of a transaction are actually performed in the second phase of the algorithm. During this phase, each replica has to release the lock (set to FREE). The cooperating algorithm is given in Algorithm 3 below.

Algorithm 3- The Cooperating Algorithm

```
TDGS_replica_process()
{
 int state;
 while (true)
 {
 //receive TDGS-RP  from primary replica
 receive (PR,b(r(p,d)));
 switch {
  case (d.lock =FREE);
     d.lock = LOCKED;
     state = SUC;
     tell the primary replica that the first phase
     returns SUC;
  case (d.lock = LOCKED)
     state =FAIL;
     tell the primary replica that the first phase
      returns FAIL;
     break;
  case (d.lock ≡ FREE ∧d.pc>0):
     d.lock = LOCKED; d.pc+=1; state = PC;
     tell the primary replica that the first phase
     returns PC;
     break;
       }
 // PR primary replica
 Receive (PR, command, NoOfPCs)
 switch {
  case (command is, "commit")
    all replicas of d will be updated
    d.lock = FREE
    // release lock
    break;
  case (command is "partial commit")
    d.lock = FREE;
    t.pc = NoOfPCs; t.b_image = beforeImage(d);
    all replicas of d will temporarily updated;
    break;
  case (command is, "abort")
     //lock to all replicas of d will be released
     d.lock = FREE;
     break;
  }
 }
```

If the network is partitioned into two disconnecting parts, they will eventually be re-united again. In this case, replicas in both partitions have partially committed updates. The conflict resolution algorithms during recovery process are responsible to make replicas in these two parts consistent. When recovering from a network partition, replicas of each part of the partition have to send a 're-uniting' message to replicas of the other part.

5. CORRECTNESS

We first assume that the life cycle of the system entities is;

Work → crash → repair and restart → work.

Without loss of generality we assume that the maximum down time is finite. For the case of failure, the recovery process will be responsible to get the entity functioning after the repair process.

Assertion 1: If a transaction returns a SUC, all its TDGS-RPs have been executed successfully.

Proof: The only way that a transaction returns a SUC is that in Algorithm 1 we have S_SUC ≠ 0. This means all TDGS-RPs to the primary replica have return SUC. The primary replica returns a SUC if and only if Algorithm 2 RSUC≠0. This means there exist a TDGS quorum of TDGS-RPs. A TDGS replica returns a SUC if and only if in Algorithm 3 the data object involved is free. If the transaction returns a SUC, in the second phase, Algorithm 1 we order the primary replica to commit the transaction. In this case, primary replica will order TDGS replicas and other replicas of the data object involved to commit in the second phase of Algorithm 2 and therefore all replicas which the data object involved will successfully perform the TDGS-RP in their second phase of Algorithm 3.

Assertion 2: If a transaction returns FAIL, no TDGS-RPs of the transaction have been executed.

Proof: The transaction returns FAIL when SFAIL≠0 in Algorithm 1. This means that the primary replica returns a FAIL. The primary replica returns a FAIL if and only if in Algorithm 2, TDGS replica returns a FAIL (all TDGS quorum cannot be formed). Furthermore, a replica returns a FAIL if and only if in Algorithm 3 it finds that the data object is not free. If the transaction returns a FAIL, in the second phase of Algorithm 1, we order the primary replica to abort the transaction. In this case, the primary replica will order the TDGS replica and other replica of the data object involved to abort in the second phase of Algorithm 2. Therefore those replicas that returned a SUC will release their locks to the data object. The replica that returned a FAIL will do nothing. In any case, none of the TDGS-RPs is executed.

6. CONCLUSIONS

A system for building reliable computing over TDGS-RP system has been described in this paper. The system combines the replication and transaction techniques and embeds these techniques into the TDGS-RP system. The replication used is discussed in Appendix. The model describes the models for replicas, TDGS-RP, transactions, and the algorithms for

managing transactions, and replicas. Finally an informal correctness analysis is carried out and various examples are described.

The paper does provide the following novel contributions: The first major contribution of the paper is the combination of the TDGS replication, transaction management, and TDGS-RP techniques to form a system that supports the development of reliable services in the TDGS-RP level. The second major contribution of the paper is the introduction of a partial commit concept to facilitate the processing of transactions when a network is partitioned. Partially committed transactions will be up-graded to a commit state or downgraded to an abort state when the partition-networks re-unite during recovery process.

REFERENCES

[1] D Agrawal and A.El Abbadi, "Using Reconfiguration For Efficient Management of Replicated Data,"*IEEE Trans. On Knowledge and Data Engineering*,vol.8,no. 5 (1996). Pp 786-801.

[2] D Agrawal and A.El Abbadi, "The generalized Tree Quorum Protocol: An Efficient Approach for Managing Replicated Data,"*ACM Trans. Database Systems*,vol.17,no. 4(1992). pp 689-717.

[3] P.A. Bernstein and N.Goodman, "An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases,"*ACM Trans. Database Systems*,vol 9,no. 4(1994),pp596-615.

[4] S.Y. Cheung, M.H. Ammar, and M. Ahmad,"The Grid Protocol: A High Performance Schema for Maintaining Replicated Data," *IEEE Trans. Knowledge and Data Engineering*,vol. 4, no. 6(1992), pp 582-592.

[5] S.M. Chung, "Enhanced Tree Quorum Algorithm For Replicated Distributed Databases," *Int'l Conference on Database* , Tokyo, 1990, pp 83-89.

[6] H. Garcia-Molina and D. Barbara,"How to Assign Votes in a Distributed System,"*J.ACM*,vol.32,no. 4(1985), pp841-860.

[7] J. Gray and A Reuter, "*Transaction Processing*", Morgan Kaufman Publishers, San Mateo, California, USA, 1993.

[8] S.Jajodia and D. Mutchles,"Dynamic Voting Algorithms for Maintaining the Consistency of a Replicated Database,"*ACM Trans. Database Systems*,vol 15,no. 2(1990), pp. 230-280.

[9] M. Maekawa," A √n Algorithm for Mutual Exclusion in Decentralized Systems,"*ACM Trans. Computer Systems*,vol. 3,no. 2(1992), pp 145-159.

[10] M. Mat Deris, A. Mamat, P.C.Seng, H. Ibrahim,"Three Dimensional Grid Structure for Efficient Access of Replicated Data", *Proceeding IEEE Int'l Conf. on Algorithms and Architectures for Parallel Processing*, World Scientific, Hong Kong, Dec. 2000 , pp.162-177

[11] M. Nocola and M. Jarke," Performance Modeling of Distributed and Replicated Databases", IEEE Trans. On Knowledge & Data Engineering, Vol.12,No.14, (2000),pp 645-670.

[12] M.T. Ozsu and P. Valduriez,"*Principles of Distributed Database Systems*", (2nd Ed., Prentice Hall, 1999).

[13] J.F. Paris and D.E. Long, "Efficient Dynamic Voting Algorithms," *Proc. Fourth IEEE Int'l Conf. Data Eng*, pp. 268-275,Feb.1988.

[14] O. Wolfson, S. Jajodia, and Y. Huang,"An Adaptive Data Replication Algorithm," *ACM Transactions on Database Systems*, vol. 22, no 2 (1997),pp 255-314.

[15] W. Zhou and A. Goscinski,"Managing Replicated Remote Procedure Call Transactions", The Computer Journal, Vol. 42, No. 7, (1999), pp 592-608.

Native ATM-support in Enhanced Communication Environment for CORBA

Xiaohong Jiang, Min Hu, Jiaoying Shi
State Key Lab of CAD&CG, Department Of Computer Science, Zhejiang University, Hangzhou, Zhejiang, 310027, P.R. China

ABSTRACT

In order to develop communication applications with high performance in heterogeneous environments, Real-time CORBA has become the focus of researches on network distributed computing. How to improve the communication performance of distributed system to meet QoS requirements is the focus in Real-time ORB system. ACE (Advanced Communication Environment) is an underlying communication framework that is used to implement the ORB end system TAO (The ACE ORB). Due to excessive processing overhead, conventional implementation of IIOP incurs very large delays, which discourage developers using CORBA in mission-critical applications. Adding ATM transport protocol into ACE is a reasonable way to improve the transmission performance. We design and implement an extension of ACE – EACE (Enhanced ATM-supported Communication Environment), upon which a QoS-support CORBA is implemented. In this paper we show the details of the implementation of EACE.

Keywords: CORBA, QoS, ACE, TAO, ATM

1. INTRODUCTION

The Common Object Request Broker Architecture (CORBA) describes the architecture of a middle-ware platform that supports portability and interoperability of applications in distributed and heterogeneous environments. The Object Management Group (OMG) issued CORBA standard in 1985, since then, many members of OMG claimed their own products according to this standard. In fact CORBA has been widely accepted as a standard of system integration framework.

First-generation CORBA middle-ware was successful in request/response applications with best-effort quality of service (QoS) requirements. Since CORBA focus on the interoperability and cooperation between object components instead of real-time limitation, conventional CORBA implementations have the disadvantages of high latency and low scalability when used for performance-sensitive applications, such as remote education, avionics automation, video on demand, and remote patient supervision. Till now, CORBA implementations have not been optimized to support real-time quality of services. Therefore CORBA implementations may not yet suite for latency-sensitive applications running over high-speed networks.[1,2]

In recent years, many research achievements have been made in real-time CORBA. In 1996 OMG issued a white paper on real-time CORBA [1], however it's only a specification and not an implementation. Sydir [3] presented a QoS-driven scheme for resource management. Halterren [4] provided IP multicast for QoS-supported CORBA based on the Open Communication Interface (OCI). Puder [5] designed and implemented the ATM Inter-ORB Protocol (ATMIOP) upon MICO, which is a free CORBA implementation developed by the international computer science institute at Berkeley.

The ACE ORB (TAO) is a free CORBA implementation developed by Washington University, which aims at system architecture for the applications with deterministic statistic real-time or best-effort QoS requirements. ACE (Advanced Communication Environment) is the communication environment of TAO. Unfortunately, it can not provide ORB binding with multiple-protocol supporting. ACE's quality of service is implemented upon the QoS service based on WinSock2. Therefore ACE can not provide QoS service on the network layer for non-WinSock systems. Some optimizations have been introduced to improve the performance of TAO. For example, the group of Pyarali [6] presented a set of optimization principles and implementation details including static and dynamic scheduling, event processing, I/O subsystem integration. However, these optimizations are incompatible with the reference model of CORBA standard and made modifications to some of the programming APIs (application programming interface). What's more, since ACE adopts TCP/IP as its transmission protocol, TCP will not be suitable for supporting critical real-time applications due to the large processing overhead.

In this paper, the design and implementation of EACE (Enhanced ACE) supporting ATM transport protocol are introduced. In section 2, we give a brief introduction to several important components of ACE. In section 3, we describe the details of implementation of EACE. At last we conclude our paper.

2. ACE FUNCTIONAL MODULES

ACE is a kernel framework for object-oriented distributed parallel mode. It is the communication environment of TAO, and it consists of reusable modules encapsulated in C++ components supporting various operating systems, such as WIN32, UNIX, and LynxOS. It has the modules for containers, concurrency, configuration, connection, IPC, memory management, name service, operation system adapters, reactor, and service configurator. The software architecture of ACE is shown in figure1 [7]. We will introduce the modules that closely relate to EACE implementation.

Container

Container is a powerful module that defines various data structures and complete common operations. It supports almost all common-used complex data structures. Through

the Container various arrays, link lists, stacks, collections or trees can be used easily. These data structures are defined as class templates, from which various data types can be derived to meet application requirements. If needed, new data types can be easily produced by class inheritance.

A *Connector* is a factory that actively establishing a connection and initializing its associated *Service Handler*. It enables creating a connection to a remote *Acceptor*.

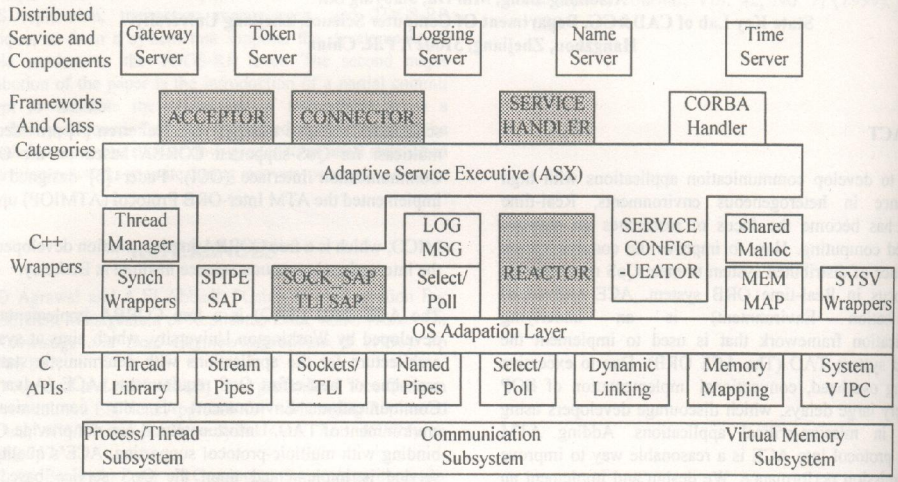


Figure 1. Component Architecture of ACE

Reactor

The Reactor provides an OO demultiplexing and dispatching framwork that simplifies the development of event-driven applications. It creates concrete event handlers by inheriting from the base class *ACE_Event_Handler*. This class specifies virtual methods that handle various types of events, such as I/O events, timer events, signals and synchronization events. Concrete event handlers are created and registered with the Reactor. When activity occurs on handlers managed by a Reactor, the Reactor automatically invokes the appropriate virtual methods on pre-registered event handlers.

The functionality of the Reactor and its registered event handlers can be extended easily without modifying or recompiling existing codes. In addition, the Reactor provides inheritance and dynamic binding to support extension of higher-level policies by applications to process events, such as connection establishment strategies, data encoding and decoding, and processing of service requests from clients.

Connection

Both *Acceptor* and *Connector* initialize the corresponding service handlers that process the data exchanged on the connection.

An *Acceptor* is a factory that implements the strategy for passively establishing a connection and initializing its associated *Service Handler*. The *Acceptor* contains a passive-mode transport endpoint factory that creates new data-mode used by *Service Handler*'s to transmit data between connected peers. *Acceptor*'s open method initializes the endpoint factory by binding it to a network address. The *Acceptor* listens to the transmission port. When a connection request arrives, the *Acceptor* creates a *Service Handler* and uses its transport endpoint factory to accept a new connection into the *Service Handler*.

IPC

IPC module provides communication service among processes. ACE provides a group of class categories based on *IPC_SAP* ("InterProcess Communication Service Access Point") base class. *IPC_SAP* encapsulates the standard I/O handle-based OS local and remote IPC mechanisms that support connection-oriented and connectionless protocols. The *SOCK_SAP* class category, which encapsulates the socket API, provides application with an object-oriented interface to the Internet-domain and UNIX-domain protocol families. The *ACE_SOCK* subclasses encapsulate Internet-domain functionality, such as *set_socket_option*, *get_local_addr*, *get_remote_addr* and *close_socket*. The class *SOCK_Acceptor* components provide connection establishment functionality typically used by servers.

OS adapter

The OS Adaptation Layer shields the higher layers of ACE from platform-specific dependencies associated with the OS mechanisms. It encapsulates OS APIs for multi-threading, multi-processing and synchronization, local and remote IPC and shared memory, synchronous and asynchronous demultiplexing for various events, explicit dynamic linking and file system APIs for manipulating files and directories.

3. IMPLEMENTATION OF EACE

In order to integrate the ATM communication protocol with CORBA we developed an ATM-supported enhanced communication environment EACE. Due to marshaling/demarshaling overhead, data copying and high levels function call overhead, conventional implementation of IIOP incur very large delays, which discourage developers using CORBA in mission-critical applications. Furthermore,

when it is concerned about stream interfaces support in CORBA, TCP transport protocol will not be suitable for time-critical applications due to its poor transmission performance. Therefore, it is a native solution to integrate ATM transfer protocol with ACE. ATM has the advantage of lower latency, good scalability, and excellent support to distributed computing systems. In addition, the programmers can create applications without the TCP/IP protocol stack and socket layer buffering mechanism.

ATM is appropriate to be used in high performance communication applications for both local wide-area networks. Using the features of defined bandwidth and quality of service on ALL5, we add ATM transport protocol in ACE without changing the ACE architecture and programming APIs.

In our enhanced communication environment EACE, multiple connections upon ATM or TCP transport protocol can be created in correspondence to various user requirements.

Transfer structure of EACE

The ATM transfer model consists of three main parts: address mapping, connection creation and data transmission. Address mapping is a basic function for network communication, which associate with a specific protocol. After the connection is established by binding with a network address (ATM address or TCP address), data transmission can be performed between connected peers. In the communication structure of ACE Connector and Acceptor perform establishing a connection and initializing data transmission. The data transmission is implemented mainly in the class of *SOCK_Stream*. The transmission has the triangle structure controlled by *Reactor*, such as shown in figure 2.

Through address mapping, the connection address is bound with a specific connection protocol(ATM or TCP) and QoS parameters, (if there exists QoS parameters). The *Acceptor* will create an instance of *SOCK_Stream*, preparing to receive the data stream. In addition, it will accept the connection sent by the communication peer and performs the function of accept.

The *Reactor* creates and registers concrete event handlers. When activity occurs on handlers managed by this *Reactor*, it automatically invokes the appropriate virtual methods on pre-registered event handlers. The control structure of *Reactor* is shown in figure 2.

Address mapping and its implementation

To implement address mapping, we defined three classes of address: *ACE_Addr, ACE_INET_Addr and ACE_ATM_Addr*. The class category *ACE_Addr* is a base address class, from which the classes *ACE_INET_Addr* and *ACE_ATM_Addr* are derived. The translation functions are implemented to translate from ACE_Addr to the two sub_classes. In addition, the virtual methods such as the operator = and! = are rewritten. Thus both the *atm_svc* address and *INET* address can be known their types through the function *get_type*. The class *ACE_ATM_Addr* is defined as follows.

```
class ACE_Export ACE_ATM_Addr:public ACE_Addr
{
 public:
    // Initialization methods.

    ACE_ATM_Addr(void);
    // default constructor
    ACE_ATM_Addr (const ACE_ATM_Addr &);
    ACE_ATM_Addr (const ACE_Addr &);
    ACE_ATM_Addr(const ASYS_TCHAR sap[]);
    //initializing address of type     ACE_ATM_Addr
    //For example:

//47.0005.80.ffe100.0000.f20f.2200.0020480694f9.00")
     ACE_ATM_Addr (const int prv_end);
    // used for local running test

    ~ACE_ATM_Addr (void);
    // default destructor

    //initializing methods after object is initialized
    Int init(void);
    //assigning address type and setting zero
    Virtual int string_to_addr(const ASYS_TCHAR
sap[]);.
    Virtual int addr_to_string(ASYS_TCHAR sap[], int
                                                    len);
    //translation between address and character string
```

It is very easy to create objects through the above five constructive functions. Among them the default constructor give a default address assigned by the system with the port address zero. The function of *ACE_ATM_addr* with the integer parameters is useful for test under the situation of uni-node running.

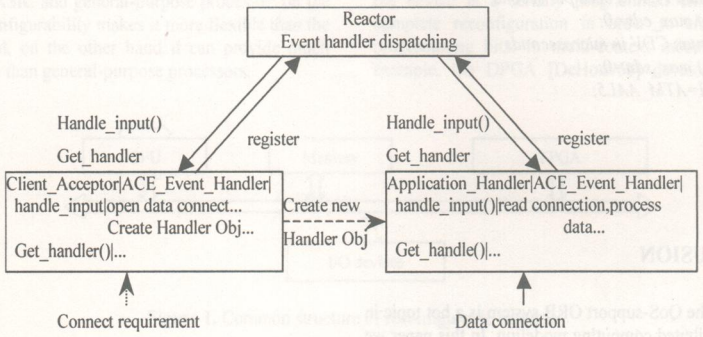


Figure 2. Transmission scheme based on Reactor event model

ATM Driver

We select the ATM driver used in Kernel 2.2.2 in Linux Mandrake 5.3, with the platform of zx-RPS switch of CABLETRON and the net card of FORE ATM.

When the socket is created, the AAL5 protocol will be used in the DGRAM way instead of STREAM. In practice, ATM protocol uses the connection-oriented communication mode. Instead of SENDTO or RECVFROM, the functions of *send* and *recv* are used to send or receive data. So its transfer differs from that of TCP/IP. We set a buffer for sending or receiving data, because in ATM driver the block transfer mode is used.

Multiple protocol support

Multiple protocol support is implemented through function overloading. It can assure a consistent programming interface using individual transfer protocols. And it also simplifies system implementation. If the connection requirement uses the ATM address, and the QoS parameters are given, a connection based on ATM protocol that meets the QoS requirements will be established. Otherwise the default TCP/IP protocol will be used.

Adding ATMQoS in EACE

The ATMQoS inherits the QoS structure from ATM driver, and it is encapsulated in the class category of *ACE_ATMQoS*. Some useful functions are also provided. The class *ACE_ATMQoS* is defined as follows.

```
class ACE_Export ACE_ATMQoS : public ATMQOS
{   // Wrapper class that holds the sender and
    // receiver flow specific information,
    // which is used by IntServ (RSVP) and
    // DiffServ.
  . public:
    int operator !=(const ACE_ATMQoS & qos) const;
    // Get/set the flow spec for data sending.

ACE_ATMQoS(void)
{
      (this->txtp).traffic_class=ATM_UBR;
         // traffic class (ATM_UBR, ...)
      (this->txtp).max_pcr=0;
         // maximum PCR in cells per second
      (this->txtp).pcr=0;
         // desired PCR in cells per second
      (this->txtp).min_pcr=0;
         // minimum PCR in cells per second
      (this->txtp).max_cdv=0;
         // maximum CDV in microseconds
      (this->txtp).max_sdu=0;
         his->aal=ATM_AAL5;
}
}
```

4. CONCLUSION

In recent years the QoS-support ORB system is a hot topic in the field of distributed computing modeling. In this paper we introduce the ACE system and show the details about how to develop EACE -- an enhanced communication environment with ATM support. Upon the platform of EACE, we implement an ORB system QTAO, which supports multiple protocol connection and QoS communication. We develop a simple experimental chat system with QTAO, in which the client can connect the server in either TCP/IP protocol or in ATM transport protocol. The test indicates that our EACE is efficient.

REFERENCES

[1] Real time SIG of OMG, "Realtime CORBA—A White Paper-Issue 1.1", Dec, 1996

[2] Lisa C. D., Roman Ginis, Russell Johnston, etc. "Expressing and Enforcing Timing Constraints in a CORBA Environment", 1966, http://citeseer.nj.nec.com/Lisa96expressing.html

[3] Sydir, J.J; Chatterjee,S.; Sabata, B. "Providing End-to-End QoS Assurances in a CORBA-Based System", Proceedings of First International Symposium on Object-Oriented Real-time Distributed Computing,(ISORC'98) 1998, pp53-61

[4] Van Halteren, A.T.; Noutash, A.; Nieuwenhuis, L.J.M. etc. "Extending CORBA with Specialised Protocols for QoS Provisioning", Proceedings of the International Symposium on Distributed Objects and Applications 1999, pp318-327.

[5] Puder A.; Moscarda M. "Native ATM support for CORBA Platforms". Proceedings of the first IEEE International Conference on ATM, 1998, pp431-438.

[6] Irfan Pyarali, Carlos O'Ryan, Douglas Schmidt, etc. "Applying Optimization Principle Patterns to Real-time ORBs", Proceedings of the 5th USENIX Conference on Object –Oriented Technologies and Systems, 1999, San Diego, CA.

[7] Douglas C. Schmidt, "An Architectural Overview of the ACE Framework", The USENIX Association Magazine on the Web, 1999.1,

[8]http://www.usenix.org/publications/login/contents/contents.jan99.htm

Design Issues in Operating Systems for RTR Systems

WU Fei and NG Kam Wing
Department of Computer Science and Engineering
The Chinese University of Hong Kong
Shatin, NT, Hong Kong
Tel: +852-26098417
Fax: +852-26035024
Email: fwu, kwng@cse.cuhk.edu.hk

ABSTRACT

Due to its potential to be able to accelerate the execution speed of a wide variety of applications greatly, reconfigurable technologies have become a hotspot in the field of computer architectures. Run-time reconfigurable (RTR) systems highlight the way to applying reconfigurable technologies to general-purpose computing. To achieve this objective, the available resources must be properly managed, thus, the OS (operating system) for a RTR system plays an important role. In this paper, we discuss several issues in the design of an OS for RTR systems.

Keywords: Reconfigurable Computing, Operating Systems,Computer Architectures

1. INTRODUCATION

Reconfigurable computing [DeHon96a][Villasenor97] is an emerging alternative to ASIC and general-purpose computing; it makes use of the reconfigurablity of some hardware devices, such as field programmable gate arrays (FPGAs) [Fanning99][Dehon96b] to construct matching circuit structures and complete computing in hardware. During the past 10 years, reconfigurable hardware (especially FPGAs) has been widely adopted and many successful application cases have proved that reconfigurable technologies are able to meet the performance requirements of many kinds of computationally intensive applications, such as image processing [Woods98], fault tolerance [Kwait95], genetic algorithms [Sidhu99], etc. Reconfigurable computing is a tradeoff between ASIC and general-purpose processor: on the one hand, the reconfigurability makes it more flexible than the ASIC method, and, on the other hand it can provide much better performance than general-purpose processors.

Computing systems that employ reconfigurable technologies are called reconfigurable systems. Most reconfigurable systems attach a FPGA board to a host computer (see Figure 1), and software programs in the host computer manage the FPGA resource. Reconfigurable systems can be classified into two types: Compile-Time Reconfigurable (CTR) systems and Run-Time Reconfigurable (RTR) systems.

Some applications running on FPGA-based systems are implemented using one single configuration per FPGA. These applications configure the FPGA before the beginning of their execution and these configurations remain active until the application is completed. Thus the functionality of the circuit does not change while the application is running. Such a system can be referred to as CTR because the entire configuration is determined at compile-time and does not change throughout the execution period. Another implementation strategy is to implement an application with multiple configurations per FPGA. In this scenario the application is divided into time-exclusive operations that need not (or cannot) operate concurrently. Each operation is implemented as a distinct configuration that can be downloaded into the FPGA as necessary at run-time. This approach is called RTR. Thus, whereas CTR applications configure the FPGA once during their execution, RTR applications typically reconfigure it many times.

RTR systems are much more flexible than CTR systems. In a RTR system, more than one computing task can share a single FPGA, with tasks larger than the FPGA size (resource) partitioned into smaller configurations to execute. Currently, most research work in the field of reconfigurable computing concentrates on RTR.

RTR systems can be classified into several models according to the different devices they adopt. In single context model, the device is a serially programmed chip that requires a complete reconfiguration in order to change any of the programming bits. In multi-context model, the device (for example, the DPGA [DeHon96b] developed at MIT) has

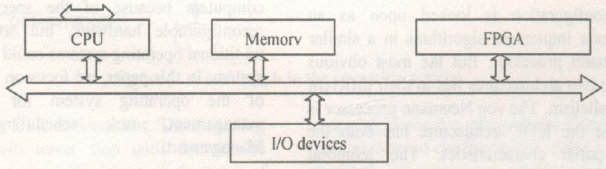


Figure 1. Common structure of reconfigurable systems

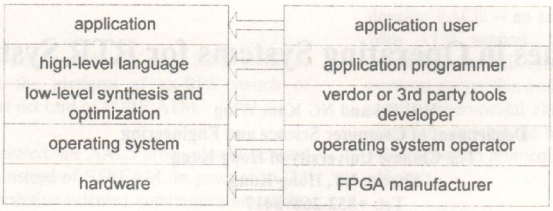


Figure 2. The similar hierarchy of the RTR system

multiple layers of configuration memory bits, each of which can be active at a different point in time. In partially reconfigurable model, devices (for example, the Virtex FPGA produced by the Xilinx) can be selectively programmed without a complete reconfiguration. RTR systems adopting partially reconfigurability meet well with the requirement of general-purpose computing. In this paper, the term "RTR system" refers to this model.

2. RTR SYSTEMS FOR GENERAL_PURPOSE COMPUTING

Till now, research institutes or commercial companies have developed quite a few reconfigurable systems. However most of these systems are aimed at solving some application-specific problems. RTR systems show people a way to solve general-purpose computing problems using reconfigurable technologies.

2.1 The von Neumann architecture vs the RTR architecture

People tend to say that in the von Neumann processor, algorithms are implemented in "software", while in reconfigurable systems, algorithms are implemented in "hardware". In a conventional processor, programs written in high-level or low-level programming languages are compiled into a series of instructions. Then the operating system loads these instructions into memory. The processor fetches instructions from memory and executes them one by one. While in reconfigurable systems, algorithms are often expressed in HDL (Hardware Description Language). Then, after high-level and low-level synthesis and optimization, one or more bit-stream files called configurations are generated. A configuration loader will download one (for example, in a non-partial RTR system) or more (for example, in a partial RTR system) configurations into the FPGA chip and start the circuit to implementing and executing the user algorithms.

Actually, if every configuration is looked upon as an instruction, RTR systems implement algorithms in a similar way to the von Neumann processor. But the most obvious distinction between the two architectures lies in their different abilities to achieve parallelism. The von Neumann processor is purely temporal, while the RTR architecture has both the temporal and the spatial characteristics. The temporal characteristic endows RTR with flexibility when implementing computation, and the spatial characteristics provides a much higher performance than a traditional processor.

2.2 Hierarchy of the RTR system

To clarify different layers of problems in constructing a RTR system for general-purpose computing, we give a hierarchy that describes the layered RTR system (see Figure 2). In this hierarchy, the left part shows different level of abstractions of the system, and the right part shows objects that are responsible for the corresponding abstraction level.

To construct a RTR system for general-purpose computing, the operating system is in a central position. It shall provide services to the high-level applications, and at the same time efficiently manage the low-level hardware resource. The functions of the operating system include mainly:

1. Managing system resources, such as FPGA chips, PCI (or other types) bus, on-chip memory.
2. Scheduling multiple user tasks to share resources of the RTR system, including: loading configuration into FPGA chips, terminating the execution of one task, temporally removing some running tasks out of FPGA chips, etc.
3. Managing communications between different configurations or between reconfigurable applications and the outside world.
4. Providing a well-developed hardware core library. Application programmers can use these hardware cores to design their own applications quickly and efficiently.

In the following part of this paper, several issues of the RTR operating system will be discussed. But since operating system study for RTR systems is currently a new field, the issues discussed here do not touch upon the high-level topics of operating systems research, and they are closely related to the basic functions of the operating system for RTR systems.

3. THE OPERATING SYSTEMS IN RTR SYSTEMS

The operating system for the RTR system is much different from the operating systems of the traditional von Neumann computers because of the special characteristics of the reconfigurable hardware, but some classical methods of traditional operating systems could be introduced into the RTR system. In this paper, we focus on several issues in the design of the operating system for RTR systems: resource management, task scheduling, and communication Management.

3.1 Resource management

The first responsibility of an operating system is to manage the

hardware resources. In a RTR system, the most important resource to be managed by the operating system is the FPGA chips. In traditional single user reconfigurable systems, the application designer can control all of the FPGA resource. Resource management in this situation is quite simple. But in a multi-user RTR system, more than one task can be implemented in the FPGA concurrently, the problem becomes much more complex. The operating system must:

1. Accommodate as many configurations as possible, which aims to make full use of available FPGA resource.
2. Protect all running tasks from being interfered by other tasks. This can ensure that the tasks are being executed accurately.

For a multi-task environment, the concept of "virtual interface" is very important. The operating system should have the ability to provide a simple virtual interface to high-level users while doing much complex administration for the low-level hardware. Since both the FPGA and the circuits are two-dimensional, data structures in the operating system shall have the ability to describe this two-dimensional characteristic. The resource management algorithms in the operating system shall be powerful enough to provide satisfactory allocation results, while still having low complexity to ensure the performance of the operating system.

3.2 Task scheduling

Here we define a task in RTR systems as a graph, with each node representing a function, and the directed edges representing the data dependencies between different functions, as illustrated in Figure 3. In a RTR system, multiple such tasks can be executed in parallel. The operating system shall schedule these tasks and coordinate communications in one task or between different tasks.

3.2.1 Task partitioning.

To download the whole task graph into the FPGA is not a wise choice. Usually, before downloading, the task graph shall be partitioned into different configurations with proper sizes. There are two methods to partition a task. One is dynamic partitioning. Dynamic partitioning can "cut" a proper sized circuit from the whole circuit according to the free space size in the FPGA, then that circuit will be downloaded into the continuous free FPGA area to execute. But since the partitioning occurs at run-time, the operating system can only afford some simple partitioning algorithms to ensure the system performance. So the result of the partitioning is usually not the most optimized scheme. It may result in heavy communication overhead between configurations. Another method is static partitioning. In this situation, partitioning is done during the synthesis phase. The user program is first synthesized into circuit pages at equal or similarly equal size (for example, each size of block is less than 10k gates). In the operating system, the FPGA chip is divided into equal sized frames (for example, one frame is 10k gates continuous FPGA space). When downloading a task into FPGA, those "ready-to-run" pages in the task are downloaded into free frames and begin to execute. Comparing to dynamic partitioning methods, the scheduling algorithm in static partitioning is much simplified, which makes the operating system more efficient. Another fact is that in the synthesis phase, nodes that have heavy communication can be synthesized into the same page to reduce the communication cost. Both the above methods are based on relocation technology, which allows the final placement of the configurations to occur at run-time.

3.2.2 Task priority

In a multi-task environment, tasks are executed concurrently. When many tasks share the limited FPGA resource, the operating system shall schedule these tasks according to their priorities, so that the FPGA can be shared efficiently and fairly, and some special tasks can be completed in an acceptable time period. In the RTR operating system, we define three types of tasks that have different priorities respectively: urgent task, real-time task and common task. Besides these external priorities, we also define internal priorities. The internal priorities can influence the execution sequence of different configurations inside a task to reduce the overall execution time of the whole task

3.2.3 Task Swapping

Swapping is one of the most significant jobs in managing a virtual hardware resource. In traditional operating systems, it is easy to stop the execution of a process, and remove the data of that process out of memory. But swapping meets some serious obstacles when being considered in a RTR environment. Tasks running in a RTR environment will never stop until the outside interruption, and it is hard to save the intermediate status of the running circuit.

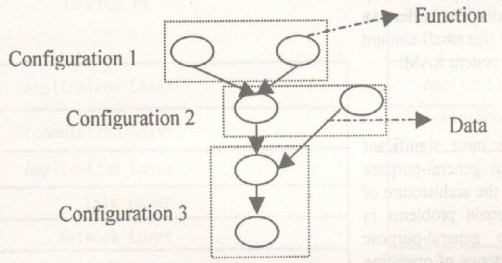


Figure 3. A task graph in the operating system

Several strategies can be adopted in implementing task swapping in the RTR operating system. The first one is swapping between configurations. Consider a running task that has a number of configurations. After a parent configuration is executed, its subsequent configurations will be downloaded into the FPGA to run. Swapping can occur at this point, that is, save the computing result of the parent configuration, remove the configuration from the FPGA, and then download the configurations of the other tasks into the free space. When the

original task is scheduled by the operating system the next time, the corresponding configuration is downloaded into FPGA, with the necessary dependent data passed into the circuit.

Another strategy is to stop a circuit crustily: in some FPGAs, there is a "stable time period" when a circuit is running. At this time, circuits can be stopped safely. But as the operating system cannot predict the status of one circuit, it must read back all the circuit's current information instead of some necessary or useful data only. While this read back action often costs much time, the efficiency of the operating system will be influenced.

3.3 Communication management

In RTR systems, the operating system must manage different types of communications, which include: the communication between configurations, the communication between FPGA chips (in a multi-FPGA environment), and the communication between configurations and the outside world. For simplicity, here only the communication between configurations will be discussed.

We will consider communications between two configurations. If the two configurations are both in the same FPGA chip, the operating system can establish a circuit to make a direct connection. The question is how to ensure that there is enough free space in a proper position to accommodate this communication circuit. To solve this problem, the operating system must reserve some free space between adjacent configurations. If the operating system adopts the run-time partitioning scheme (as described in the foregoing part of the paper), it must put the partition in a larger free space so as to reserve space to route the communication. If the static partitioning is adopted, additional space is also needed when a circuit page is downloaded into a FPGA frame.

If the communication occurs between two configurations but only one of them is in the FPGA, the operating system shall have the ability to temporarily save the communication data. When the corresponding configuration is loaded into the FPGA chip, the operating system restores the communication data and sends it to the proper position. This kind of buffered communication is especially important. Actually it is not only used for off-chip communication, but also used for on-chip communication between asynchronous circuits. The buffer can either be placed in the FPGA on-chip RAM (for small amount of data) or in the onboard RAM or even the system RAM.

4. CONCLUSION

Operating system research is one of the most significant subjects in applying RTR technologies to general-purpose computing. In this paper, we first analyzed the architecture of current RTR systems, discussed several main problems in applying reconfigurable technologies to general-purpose computing. Then we emphasized the importance of operating systems and gave a detailed description of features that a RTR operating system should have. For some issues related to the functionality of the RTR operating system, we gave general analysis and possible solutions. The issues discussed here are mainly about the functionality, and they cover only a small part of the overall design of the operating system. Besides these low-level topics, to design an operating system for RTR systems requires much more studies on system flexibility, system security, etc. We are on the way to combine traditional operating system's design methods and the characteristics of reconfigurable hardware together to solve these problems.

REFERENCES

[Dehon96a] Andre Dehon, Role of Reconfigurable Computing, Panel Presentation for reconfig.com RC Roundtable, Oct 1996.

[Dehon96b] Andre Dehon, DPGA Utilization and Applicatio ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 1996

[Fanning99] J Fanning, FPGA's Literature Review http://dias.umist.ac.uk/ NJG/fpga2.htm.

[Kwiat95] Kwait, K.A.; and Hariri; S.; Efficient Hardware Fault Tolerance Using Field-Programmable Gate Arrays, Proceedings ISSAT International Conferenc on Reliability and Quality in Design, 1995.

[Sidhu99] R. P. Sidhu, A. Mei, and V. K. Prasanna. Genetic Programming using Self-Reconfigurable FPGAs. In International Workshop on Field Programmable Logic and Applications, September 1999.

[Steve00] Steve Guccione, List of FPGA-based Computing Machines, http://www.io.com/~guccione/HW_list.html Last updated: 21 August 2000.

[Villasenor97] John Villasenor and William H. Mangione-Smith, Configurable Computing, Scientific American, June 1997.

[Woods98] R.F. Woods, D. Trainor and J_P Heron, Applying an XC6200 to real-time Image Processing, IEEE Design & Test of Computers, January-March 1998.

[Xilinx01] Xilinx, Inc. Virtex II Handbook, 2001.

Response Time of Urgent Aperiodic Message In Foundation Fieldbus*

Zhi Wang[1] Youxian Sun[1], Tianran Wang[2]
[1] National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027, China
[2] Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110015, China
Author of correspondence (email: wangzhi@iipc.zju.edu.cn)

ABSTRACT

This paper derives formula for the response time of periodic messages in Foundation Fieldbus (FF). Authors first consider periodic messages of different criticality levels in timing requirement, such as urgent and normal, and derive formula for response time of these messages. Then through analyzing the formula, authors find the deficiency of PT counting mechanism, and propose a new PT counting mechanism to enhance realtime response of urgent periodic messages. In the end, the advantage of the new counting mechanism is validated by numerical instance.

Keywords: Foundation Fieldbus, Real-time, Periodic Traffic, Field Communication, Response Time, Token Circulation Period

1. INTRODUCTION

Distributed computing and applications (DCA) plays an important role in many areas, such as electronics commerce, engineering design, science computing etc. One of popular DCA is distributed realtime system (DRS), which is widely applied in industrial process control, automation manufacturing etc. DRS is characterized by its timely execution of tasks that usually reside on different nodes and communicate with one another to accomplish a common goal. End to end deadline guarantee of these tasks are possible only if a communication network support timely delivery of inter-task messages [1-3].

Fieldbus, as infrastructure of communication to support realtime traffic of field devices in factory floor, is generally characterized by the obligation to respect stringent temporal constraints that must be met in order to guarantee correctness and safe. Consequently, a significant issue is devoted to taking account of temporal property of fieldbus to guarantee timing requirement of field devices. Therefore, scholars, such as Burns, Shin, Song and Tover, pay lots of attentions on evaluation and analysis of temporal property within many widely applied fieldbuses, such as CAN、PROFIBUS、P-NET and WorldFIP [4-15].

FF is one of the most popular fieldbus standards. FF not only provides complete control function through specifying user layer protocol, including Function Block (FB) and Device Description (DD), and explicitly distinguishes traffics into periodic and periodic and manages them with different respective strategies [4]. However, FF lacks accurate temporal property in periodic traffics, particularly its response time, which is the main focus of this study. This study addresses important issues related to guaranteeing periodic message, such as priority of periodic traffics, and priority and PT cycle period, and worst-case response time.

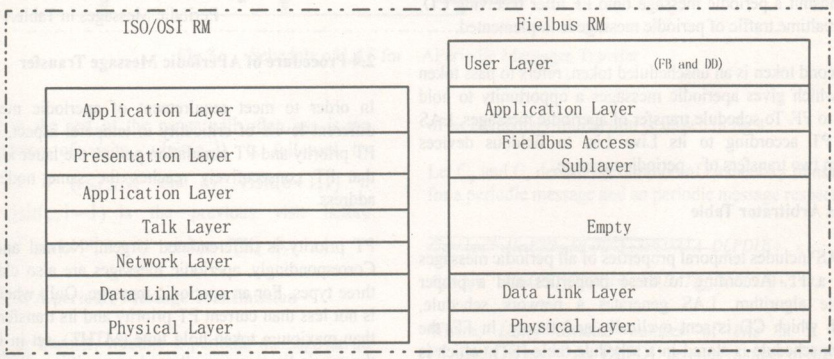


Fig.1. The Relationship between FF and ISO/OSI RM

* This work is supported by NSFC (60084001)

2. FOUNDATION FIELDBUS

2.1 The Architecture of FF

Considering the real time requirement of factory floor, FF adapts optimized ISO/OSI Reference Model (RM), and only reserves physical layer, data link layer (DLL) and application layer. In addition, FF provides the user layer, which defines a standard of user applications. Through specifying FB and DD, the user layer provides all necessary for control function and information needed for a control system or host to understand the meaning of data. FB and DD, which meet critical requirement of a control system in control function, interoperability and openness, make FF not only a communication standard, but also a control system standard. The relationship between FF four layers architecture and ISO/OSI RM is shown in Fig.1.

2.2 Messages and Token

Because of the importance of realtime traffic in factory floor, and of the variety of messages in temporal characteristics, FF explicitly distinguishes periodic and periodic messages, and applies different strategies to support these messages. For guaranteeing temporal constraints of its applications, FF applies centralized media access mechanism based on a producer/ distributor/consumer (PDC) model, which relates transmitter and receiver within a distributed system. In PDC model, there is an arbitrator, which is a centralized bus controller for regulating communication and interoperation of messages between producers and consumers.

FF implements the arbitrator using Link Active Schedule (LAS), which manages schedule of all messages in FF through broadcasting token. Any fieldbus devices can only transmit its waiting messages after receiving token. For effectively managing periodic and periodic message, which are characterized with their temporal property, FF provides two types of tokens.

The first one is a scheduled token, refers to compel data (CD), which only support periodic messages. According to its stored predefined time, LAS broadcasts CD. CD includes a specific address, through which the node corresponding the address will transmit a periodic message onto FF after receiving CD. Thus, realtime traffic of periodic message is implemented.

The second token is an unscheduled token, refers to pass token (PT), which gives aperiodic messages a opportunity to hold access to FF. To schedule transfer of aperiodic messages, LAS sends PT according to its Live List to fieldbus devices between two transfers of periodic message.

2.3 Bus Arbitrator Table

The LAS includes temporal properties of all periodic messages within a FF. According to these properties and a proper schedule algorithm, LAS generates a network schedule, through which CD is sent cyclically and timely. In FF, the network schedule is stored in a schedule table (ST), which is made up of a set of basic schedule table, known as micro cycles.

The LAS gives access right to periodic messages through sending CD according to its schedule stored in each micro cycle within the ST, then gives access right to aperiodic message through sending PT if time left in this micro cycle is enough.

The portion of a micro cycle reserved for periodic message is denoted as periodic window, whereas the time left after the periodic window is denoted as aperiodic window. The transfers for two types of messages depend on schedules stored in their respective windows. Once all microcycles have been performed in the ST, LAS repeats the same network schedule from the beginning.

Two important parameters are associated with the ST, micro cycle and macro cycle. The former imposes the maximum rate at which the BA performs a set of scans, and the latter is minimum duration during which the sequence of microcycles is repeated. Usually, the microcycle and the macrocycle are respectively set equal to the HCF and the LCM of the required scan periodicities.

Table.1 Example set of periodic messages

Variable Identifier	A	B	C	D	E	F
Periodicity (ms)	1	2	3	4	4	6

Let N_{Mic} denote number of micro cycles within ST. If ST is constructed by rule of HCF and LCM, N_{Mic} equals to LCM/HCF.

Assume that within a FF there are 6 periodic messages with arrival rate in Table.1 to be transferred. According rule of HCF and LCM, the microcycle and macrocycle are set to 1ms and 12 ms respectively. A feasible schedule meeting realtime requirements of these periodic messages is illustrated as Fig.2 where we consider C_p=0.2ms for communication time of each periodic message.

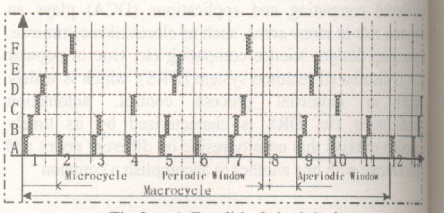


Fig.2. A Feasible Schedule for Periodic Messages in Table.1

2.4 Procedure of APeriodic Message Transfer

In order to meet requirement of aperiodic messages with different levels of criticality in temporal aspect, FF provides PT priority and PT circulation period, the latter is the duration that PT consecutively reaches the same node with least address.

PT priority is differentiated Urgent, Normal and Available. Correspondingly, aperiodic messages are also classified with three types. For an aperiodic message, Only when its priority is not less than current PT priority and its transfer time is less than maximum token hold time (MTHT), set in PT frame, is the aperiodic message allowed transmitting. PT returns LAS when no aperiodic message with proper priority exists or MTHT expires.

PT circulation period includes setting PT circulation period (SCT) and actual PT circulation period (ACT), where the former is set online or offline by an operator and the latter is

measured online. To enhance realtime response of critical aperiodic message and to adapt load in FF, LAS online changes PT priority according to the difference of between SCT and ACT. ACT being larger than SCT, LAS increases PT priority unless PT priority is urgent.

Otherwise, LAS decreases PT priority unless PT priority is available. Thus, FF as far as possible meets timing requirement of all aperiodic messages through setting PT priority, SCT and ACT. Besides, in order to decrease the fluctuation of PT priority, FF provides difference counter (DC) by which PT priority is adjusted automatically. In this paper, DC is designed 1. The LAS schedule for aperiodic message transfers is shown in fig.3.

3. MODEL OF APERIODIC MESSAGE COMMUNICATION

3.1 Model of PT Transmission

Let N be the number of nodes in FF and assume that one FF connection per node. Additional notation needs to be described is a particular behavior of sequence which describes number of PT cycle and number of node PT visiting. PT cycle refers to duration that PT visits nodes from the least address to the last address. We will index PT visiting a node by a pair of subscripts, visit(c, i), where c indicates the number of PT cycles, and i indicates the node being visited.

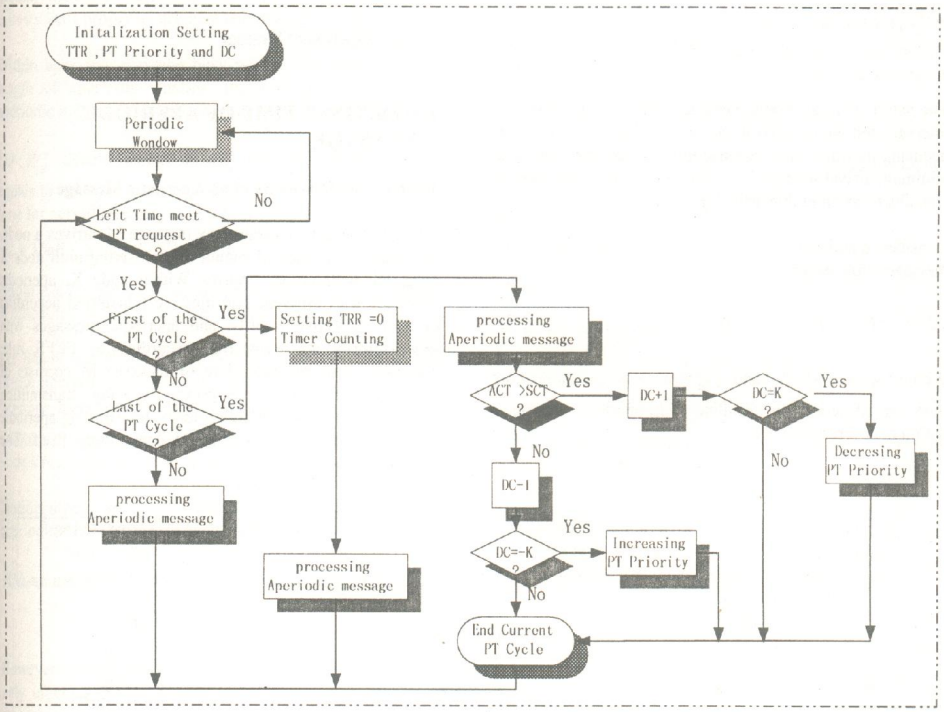


Fig.3. Schedule of LAS for APeriodic Messages Transfer

The notion in the rest of the paper will often use is the natural ordering of visits. visit(c, i) is followed by visit(c,i+1) if $(1 \le i \le N)$ and by visit(c+1,1) if $(i=N)$. visit(c, i−1) is the previous visit before visit(c, i) if $(i \ne 1)$, otherwise visit(c−1,N) is.

3.2 Model of A periodic Message Transmission

For a message transfer, whether a periodic message or an aperiodic message, only needs two consecutive phases, getting access to FF and transmitting message, the message transfer hence constructs an elementary transaction. During a normal operation, an elementary transaction at least needs two frames, CD (PT) frame and Data frame. The procedure of an elementary transaction is shown in fig.4.

Let C_p and C_a denote the duration of elementary transactions for a periodic message and an periodic message respectively.

$$C_p = bps^{-1} \cdot (len(CD\_DLPDU) + len(DATA\_DLPDU)) + \times t_r \quad (1)$$

$$C_a = bps^{-1} \cdot (len(PT\_DLPDU) + len(DATA\_DLPDU)) + 2 \times t_r \quad (2)$$

Where, *Len*() denotes number of *bits* in a frame, PT_DLPDU and Data_DLPDU denotes number of *bits* in PT and a data frame respectively (10octets, 22octets), t_r denotes the minimum duration between the transmission of two frames (set in configure phase, normally is 12octets). Within H1 fieldbus, *bps*=31.25 *bits/s*, therefore, the transmission of an aperiodic message is approximately 14-20ms [4].

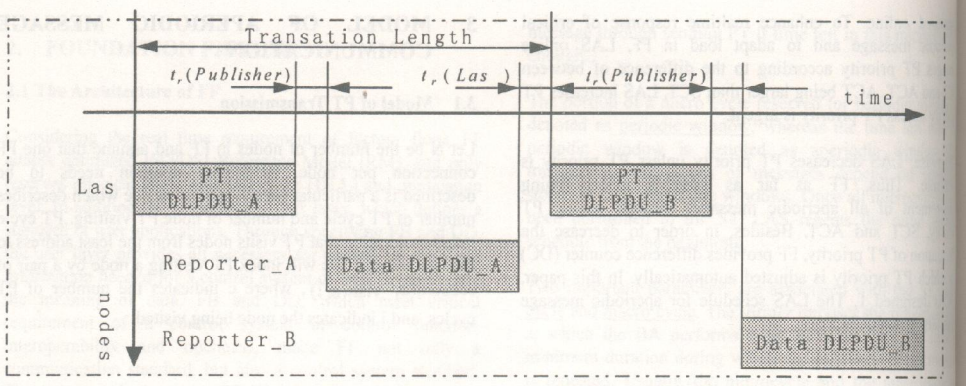


Fig 4. Elementary Transaction for Periodic (Aperiodic) Message

The nature of an aperiodic message is in its irregular arrival interval, and in no arrival period existing. However, for deducing its worst-case response time, we assume there is a minimum arrival interval for it, and assign an arrival period equaling to its minimum arrival interval.

Considers a node k (k=1, … , N), in which a set of aperiodic messages Ma^k exists.

$$Ma^k = (Ca_i^k, Ta_i^k, Da_i^k)\ (i = 1 \cdots np) \tag{3}$$

Within Eq(3), Ca_i^k, Ta_i^k and Da_i^k represents communication time, arrival period and deadline of i_{th} aperiodic messages within node k respectively.

4. RESPONSE TIME OF APERIODIC MESSAGE

4.1 Response Procedure of an Aperiodic Message

When an aperiodic message, say message M, arrives a node, say node X, message M cannot be transferred until node X owns PT with proper priority. Within node X, aperiodic messages with different priorities are transferred according to their respective priority, and aperiodic messages with same priority are transferred according to FCFS rule. Therefore, response procedure of message M consists of arrival of PT with proper priority arriving, transmitting aperiodic messages with high priority, transmitting aperiodic messages with same priority, and transmitting itself. The response procedure is shown in fig 5.

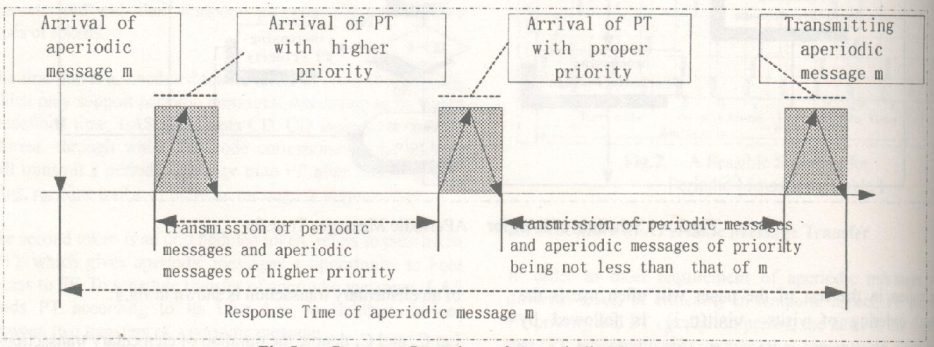


Fig 5. Response Procedure of an Apriodic Message M

4.2 Busy Period of an APeriodic Message

The above analysis indicates that response procedure of an aperiodic message is a complicate procedure since it is concerned with ST, PT priority, aperiodic message priority, SCT and ACT.

For guaranteeing real time requirement of an aperiodic message, we must know its response time under various kinds of situation. However, to deduce the response time of an aperiodic message is difficult if it is possible. Fortunately, we can determine that the worst-case situation for an aperiodic message is in its busy period. First we introduce concept of critical instant, then concepts of various kinds of busy periods.

【Definition1】 Critical instant is an instant all messages arrive simultaneously and suffer maximum response time.

【Definition2】 Apriodic window busy period is a duration, beginning from critical instant until no aperiodic message existing, during which aperiodic window will be always busy.

【Definition3】 Urgent aperiodic window busy period is a duration under PT priority being urgent at critical instant, which begins from critical instant until no urgent aperiodic message existing, and during which aperiodic window only transmits urgent aperiodic messages.

【Definition4】 Normal aperiodic window busy period is a duration under PT priority being normal at critical instant, which begins from critical instant until no urgent aperiodic message existing, and during which aperiodic window not only transmits urgent periodic messages but also normal aperiodic messages.

Similarly, we can define available aperiodic window busy period, which is omitted in this paper since available PT priority isn't proper to stringent temporal requirement.

Within aperiodic message busy period, ST determines the length of aperiodic window; therefore response time of aperiodic messages is function of periodic message schedule.

Let W_a^i denote length of i_{tha} aperiodic window, which equals to left time of i_{th} microcycle after subtracting transfer time for periodic messages in this microcycle.

$$W_a^i = MicP - \sum_{j=1}^{np} (table[j, i^*]) \times Cp_j \tag{4}$$

within Eq(7), $table[j, i^*]$ denotes whether periodic message j is scheduled in i_{th}^* microcycle, $i^*=[(i\text{-}1)\ Mod\ N_{Mic}]+1$.

Let $BNum$ and $BNnm$ denote number of microcycles within urgent aperiodic window busy period and normal aperiodic window busy period respectively, na_k^u and na_k^n denotes number of urgent and normal aperiodic messages within node k respectively.

During urgent aperiodic window busy period only urgent aperiodic messages are transmitted, therefore,

$$BNum = min(\psi) \bigcap \sum_{i=1}^{\psi} W_a^i \geq \sum_{k=1}^{N} \sum_{j=1}^{na_k^u} \left(Ca_k^{uj} + \left\lfloor \frac{\psi \times MicP}{Ta_k^{uj}} \right\rfloor \cdot Ca_k^{uj} \right) \tag{5}$$

However, during normal aperiodic window busy period not only urgent aperiodic messages but also normal aperiodic messages are transmitted, except the last PT visit during which transfer time of normal aperiodic messages is irrespective, so,

$$BNnm = \min(\xi) \bigcap \sum_{i=1}^{\xi} W_a^i \geq \sum_{k=1}^{N} \sum_{j}^{na_k^n} \left(Ca_k^{uj} + Ca_k^{nj} + \left\lfloor \frac{\xi \times MicP}{Ta_k^{nj}} \right\rfloor \cdot Ca_k^{nj} + \left\lfloor \frac{\xi \times MicP}{Ta_{k-1}^{nj}} \right\rfloor \cdot Ca_{k-1}^{nj} \right) \tag{6}$$

4.2 Maximum Response Time of Urgent APeriodic Messages

An urgent aperiodic message suffers worst-case response time within aperiodic window busy period, hence during which calculations for its response time is sufficient. Response time of an urgent aperiodic message being less than its deadline indicates that the urgent aperiodic message meets its temporal requirement, otherwise the urgent aperiodic message does not and its calculation stops. We separately deduce response times of urgent aperiodic messages under two cases, PT priority being urgent at critical instant and PT priority being normal, since PT priority can change to normal except SCT is set as small as 0.

4.2.1 Analysis of Response Time under PT Priority Being Urgent at Critical Instant

Under PT priority being urgent at critical instant, response time of an urgent aperiodic message in node k includes waiting PT arriving to node k (only urgent aperiodic messages are transmitted during the period) and waiting other urgent aperiodic messages arrived earlier in node k.

In this case, Let $MNum(y,x)$ and $MNnm(y,x)$ denote number of micro cycles from $visit(1,1)$ to $visit(y,x)$ for transmitting urgent and normal aperiodic messages in node x respectively. Let $Ra(y,x)$ denote response time of urgent aperiodic messages during $visit(y,x)$.

Before completing $visit(1,1)$, FF must transmit urgent aperiodic messages arrived in critical instants and new arrival during waiting period. So,

$$MNum(1,1) = \min(\psi) \bigcap \sum_{i=1}^{\psi} W_a^i \geq \sum_{j=1}^{na_1^u} \left(Ca_1^j + \left\lfloor \frac{\psi \times MicP}{Ta_1^{uj}} \right\rfloor \cdot Ca_1^{uj} \right) \tag{7}$$

$$Ra(1,1) = (MNum(1,1) - 1) \times MicP + \sum_{j=1}^{np} table\left[j, MNum(1,1)^*\right] \times Cp_j + \sum_{j=1}^{na_1^u} Ca_1^{uj} \tag{8}$$

Similarity, we can deduce response times of urgent aperiodic messages in any node, say node m.

$$MNum(1,m) = \min(\psi) \bigcap \sum_{i=1}^{\psi} W_a^i \geq \sum_{j=1}^{na_m^u} \left\lfloor \frac{\psi \times MicP}{Ta_m^{uj}} \right\rfloor \cdot Ca_m^{uj} + \sum_{k=1}^{m} \sum_{j=1}^{na_1^u} Ca_k^{uj} + \sum_{k=1}^{m-1} \sum_{j=1}^{na_m^u} \left\lfloor \frac{MNum_1^k \times MicP}{Ta_k^{uj}} \right\rfloor \cdot Ca_k^{uj} \tag{9}$$

$$Ra(1,m) = (MNum(1,m) - 1) \times MicP + \sum_{j=1}^{np} table\left[j, MNum(1,m)^*\right] \times Cp_j + \sum_{j=1}^{na_m^u} Ca_m^{uj} \tag{10}$$

In the same way, we can deduce response times of urgent aperiodic messages during any PT cycle, that means in any $visit(y,x)$ ($x \geq 2$).

$$MNum(y,x) = \min(\psi) \bigcap \sum_{i=1}^{\psi} W_a^i \geq \sum_{j=1}^{na_y^u} \left\lfloor \frac{\psi \times MicP}{Ta_y^{uj}} \right\rfloor \cdot Ca_y^{uj} + \sum_{k=1}^{N} \sum_{j=1}^{na_1^u} Ca_k^j + \sum_{u,v=x-1,y+1}^{x,y-1} \sum_{j=1}^{na_1^u} \left\lfloor \frac{MNum_u^v \times MicP}{Ta_v^{uj}} \right\rfloor \cdot Ca_v^{uj} \tag{11}$$

$$Ra(y,x) = (MNum(y,x) - 1) \times MicP + \sum_{j}^{np} table\left[j, MNum(y,x)^*\right] \times Cp_j + \sum_{j}^{na_m^u} Ca_m^{uj} \tag{12}$$

the above calculation continues until one of the two cases occur, $MNum(y,x)$ being not less than $BNum$ or response time an urgent aperiodic message being large than its

deadline. The former case indicates real time requirement of urgent aperiodic messages are met, and the latter case are not.

4.2.2 Analysis of Response Time under PT Priority Being Normal at Critical Instant

Under PT priority being normal at critical instant, response time of an urgent aperiodic message in node k includes waiting arrival of PT to node k (urgent and normal aperiodic messages are transmitted during the period) and waiting of other urgent aperiodic messages arrived earlier in node *k*.

In this case, let $\tilde{R}a(y,x)$ denote response time of urgent aperiodic messages in node x during $\text{visit}(y,x)$, let $\tilde{M}Num(y,x)$ and $\tilde{M}Nnm(y,x)$ denote number of microcycles from $\text{visit}(1,1)$ to $\text{visit}(y,x)$ for transmitting urgent and normal messages respectively.

Because in the same node urgent aperiodic messages are transmitted first, therefore,

$$\tilde{M}Num(1,1) = MNum(1,1) \tag{13}$$

$$\tilde{R}a(1,1) = (\tilde{M}Nnm(1,1)-1)\times MicP + \sum_{j=1}^{np} table\left[j, MNnm(1,1)^*\right]\times Cp_j + \sum_{j=1}^{na_1^u} Ca_1^{uj} \tag{14}$$

For urgent aperiodic messages in node m, they must wait PT returning, and during the period FF transmits urgent and normal aperiodic messages in node x before completing $\text{visit}(1,x)$ (x<m). Besides, FF must transmit urgent aperiodic messages arriving node m before completing $\text{visit}(1,m)$.

$$\tilde{M}Num(1,m) = \min(\psi)\cap\sum_{i=1}^{\psi} W_a^i \ge \sum_{k=1}^{m}\sum_{j=1}^{na_k^u} Cd_k^{uj} + \sum_{j=1}^{nd_m^u}\left\lfloor\frac{\psi\times MicP}{Td_m^{uj}}\right\rfloor\cdot Cd_m^{uj} + \sum_{k=1}^{m-1}\left(\sum_{j=1}^{nd_k^u}\left\lfloor\frac{\tilde{M}Nun(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj} + \sum_{j=1}^{nd_k^n}\left\lfloor\frac{\tilde{M}Nnm(1,k)\times MicP}{Td_k^{nj}}\right\rfloor\cdot Cd_k^{nj}\right) \tag{15}$$

$$\tilde{M}Nnm(1,m) = \min(\psi)\cap\sum_{i=1}^{\psi} W_a^i \ge \sum_{k=1}^{m}\left(\sum_{j=1}^{nd_1^u} Cd_k^{uj} + \sum_{j=1}^{nd_1^n} Cd_{k-1}^{nj}\right) + \sum_{j=1}^{nd_m^u}\left\lfloor\frac{\psi\times MicP}{Td_m^{uj}}\right\rfloor\cdot Cd_m^{uj} + \sum_{k=1}^{m}\sum_{j=1}^{nd_m^u}\left\lfloor\frac{MNun(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj} + \sum_{k=1}^{m-1}\sum_{j=1}^{nd_k^u}\left\lfloor\frac{MNnm(1,k)\times MicP}{Td_k^{nj}}\right\rfloor\cdot Cd_k^{nj} \tag{16}$$

$$\tilde{R}a(1,m) = (\tilde{M}Nnm(1,m)-1)\times MicP + \sum_{j=1}^{np} table\left[j, \tilde{M}Nnm(1,m)^*\right]\times Cp_j + \sum_{j=1}^{na_m^u} Ca_m^{uj} \tag{17}$$

The above calculation continues until one of the two cases occur, $\tilde{M}Num(y,x)$ being not less than $BNnm$ or response time of an urgent aperiodic message being large than its deadline. The former case indicates real time requirement of urgent aperiodic messages are meet, and the latter case are not.

4.3 The Improvement of ACT Counting Mechanism

Eq(15)~(17) indicates that response time of urgent aperiodic messages is badly enlarged, and that can't be avoided because PT priority can change to urgent only when the current PT cycle ends and ACT being less than SCT. Setting less SCT can maintain PT priority at urgent, however that maybe make normal aperiodic messages lose chance of transmission.

Therefore it is necessary to propose a new one to enhance real-time traffic of FF for urgent aperiodic messages. It is obvious that the new one should respond real time requirement of urgent aperiodic messages and load in FF. This paper introduces a mechanism in which ACT equals the elapsed time between PT twice consecutively arriving to the same node X. X is not confined to 1.

Aperiodic window busy period under the new mechanism apparently can't exceed the larger between the two previous introduced aperiodic window busy periods, so it is omitted here.

In this new ACT counting mechanism, let $\tilde{M}Num(y,x)$ and $\tilde{M}Nnm(y,x)$ denote number of micro cycles from $\text{visit}(1,1)$ to $\text{visit}(y,x)$ for transmitting urgent and normal aperiodic messages in node x respectively. Let $\tilde{R}a(y,x)$ denote response time of urgent aperiodic messages during $\text{visit}(y,x)$.

Assume PT priority is normal at critical instants and ACT of node m is larger than SCT, then PT priority will increase to urgent at the beginning of $\text{visit}(1,m)$.

$$\tilde{M}Num(1,m) = \min(\psi)\cap\sum_{i=1}^{\psi} W_a^i \ge \sum_{k=1}^{m-1}\sum_{j=1}^{nd_k^u}\left(Cd_k^{uj} + \left\lfloor\frac{\tilde{M}Nun(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj}\right) + \sum_{k=1}^{m-1}\sum_{j=1}^{nd_k^n}\left(Cd_k^{uj} + \left\lfloor\frac{\tilde{M}Nnm(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj}\right) + \sum_{j=1}^{nd_m^u}\left(Cd_m^{uj} + \left\lfloor\frac{\psi\times MicP}{Td_m^{uj}}\right\rfloor\cdot Cd_m^{uj}\right) \tag{18}$$

$$\tilde{R}a(1,m) = (\tilde{M}Num(1,m)-1)\times MicP + \sum_{j=1}^{np} table\left[j, \tilde{M}Num(1,m)^*\right]\times Cp_j + \sum_{j=1}^{na_m^u} Ca_m^{uj} \tag{19}$$

Under the new ACT counting mechanism, PT priority changes frequently, it is not necessary and difficult to deduce response time of urgent aperiodic messages in any pattern of PT priority. Here, only a special case is studied. If ACT of node 1 is larger than SCT until $\text{visit}(y,x)$ ($\text{visit}(y,x)\ge\text{visit}(1,m)$), then PT priority will maintain at urgent until ending of $\text{visit}(y,x)$ $\text{visit}(1,m)$.

$$\tilde{M}Num(y,x) = \min(\psi)\cap\sum_{i=1}^{\psi} W_a^i \ge \sum_{j=1}^{na_x^u}\left\lfloor\frac{\psi\times MicP}{Td_x^{uj}}\right\rfloor\cdot Cd_x^{uj} + \sum_{u,n=y-1,x+1}^{y,x}\sum_{j=1}^{na_x^u}\left\lfloor\frac{\psi\times MicP}{Td_x^{uj}}\right\rfloor\cdot Cd_x^{uj} + \sum_{k=1}^{m-1}\left(\sum_{j=1}^{na_k^u}\left(Cd_k^{uj} + \left\lfloor\frac{\tilde{M}Num(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj}\right) + \sum_{j=1}^{na_k^n}\left(Cd_k^{uj} + \left\lfloor\frac{\tilde{M}Nnm(1,k)\times MicP}{Td_k^{uj}}\right\rfloor\cdot Cd_k^{uj}\right)\right) \tag{20}$$

$$\tilde{R}a(y,x) = (\tilde{M}Num(y,x)-1)\times MicP + \sum_{j=1}^{np} table\left[j, \tilde{M}Num(y,x)^*\right]\times Cp_j + \sum_{j=1}^{na_m^u} Ca_m^{uj} \tag{21}$$

The difference between $\tilde{M}Num(y,x)$ and $\tilde{M}Nnm(y,x)$ explicitly indicates that real time response of urgent aperiodic message is enhanced by the new ACT counting mechanism.

5. NUMBERICAL EXAMPLES

Consider a FF scenario with 4 nodes, each one with the following message streams.

Table 1. Numerical Example

Node 1	Node 2	Node 3	Node4
$Ca_1^{n1} = 8ms$	$Ca_2^{n1} = 9ms$	$Ca_3^{n1} = 8ms$	$Ca_4^{n1} = 5ms$
$Ca_1^{n2} = 6ms$	$Ca_2^{n2} = 7ms$	$Ca_3^{n1} = 18ms$	$Ca_4^{n2} = 6ms$
$Ca_1^{n3} = 9ms$	$Ca_2^{n1} = 15ms$	$Ca_3^{n2} = 24ms$	$Ca_4^{n1} = 22ms$
$Ca_1^{n1} = 15ms$	$Ca_2^{n2} = 25ms$	$Ca_1^{n3} = 30ms$	

Table 2. Response Time of Urgent Aperiodic message

$SCT = 90ms$ $BNUM = 67ms$	Node 1	Node 2	Node 3	Node4
$BNNM = 67ms$	$\bar{R}a(1,1) = 14ms$	$\bar{R}a(1,1) = 54ms$	$\bar{R}a(1,3) = 102ms$	$\bar{R}a(1,4) = 175ms$
	$\tilde{R}a(1,1) = 14ms$	$\tilde{R}a(1,1) = 54ms$	$\tilde{R}a(1,3) = 102ms$	$\tilde{R}a(1,4) = 113ms$

It is obvious that SCT can be set as small as *0*, however if this is the case, normal aperiodic messages would not be transferred at all. Therefore, the new ACT counting mechanism is feasible approach for not only guaranteeing realtime traffic of urgent aperiodic messages, but also making the best of transmitting normal aperiodic messages. The feasibility of the new ACT counting mechanism is validated through the example.

6. CONCLUSION

FF communication mechanism and its effect on guaranteeing real time traffic of critical aperiodic messages are analyzed, and a new mechanism is proposed to enhance its realtime capability. First an integrated message transmission model, which integrates aperiodic messages and aperiodic messages together, is established through analyzing LAS schedule. Then formula for response times of urgent aperiodic messages are given after considering the effect of SB, ACT, SCT and PT priority on aperiodic messages transmission. Further, deficiency of ACT counting mechanism in meeting realtime traffic of urgent aperiodic messages is found through the formula, and a new one is proposed consequently to improve the deficiency. The advantage of the new one is validated by numerical instance in the end. The ongoing work is to set SCT and optimize aperiodic messages priorities under some criterion, such as response time, loss rate, utilization, etc.

REFERENCES

[1] H. Kopetz, Real Time System Design Principles for Distributed Embedded Application, *Kluwer Academic Publishers*, 1997

[2] C. Buttazzo, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Application, *Kluwer Academic Publisher*, 1997

[3] Foundation Specification System Architecture Austin, 1996

[4] Jean Thomesse, The Fieldbus, *International Conference of Intelligent Components and Instruments for Control Application*, 1997, 13-23

[5] S. Cavalieri, Impact of Fieldbus on Communication in Robotic Systems, *IEEE Trans on Robot and Automation*,13(1), 30-48, 1997.

[6] K. Tindell and A. Burns, Analysis of Hard Real-Time Communication, *Real-Time Systems*, 9(2), 147-173, 1997

[7] K.M. Zuberi and K G. Shin, Design and Implementation of Efficient Message Scheduling for CAN, *IEEE Trans. on Computers*, 49(2), 182-188, 2000

[8] N. Navet, Y.Q Song, and F. Simonot, Worst-Case Deadline Failure Probability in Real-Time Applications over CAN, *Journal of Systems Architecture*, 46(7), 607-617, 2000

[9] W. Zhao and J. Stankovic, A Window Protocol for Transmission of Time Constrained Messages, *IEEE Transactions on Computers*, 39(9), 1186-1203, 1990

[10] Y. Kim and S. Jeong, Pre-Run-Time Scheduling Method for Distributed Real-Time Systems in a FIP Environment, *Control Engineer Practice*, 6(1) 103-109

[11] E. Tovar and F. Vasques, Real-Time Fieldbus Communi- cations Using Profibus Networks, *IEEE Transactions on Industrial Electronics*, Vol.46(6), pp. 1241-1251, 1999

[12] E. Tover and F. Vasques, Supporting Realtime Distributed Computer Control System with Multi-hop P-Net Network, *Control Engineering Practice*, 7(11), 1015-1025, 1999.

[13] C.C. Chou, K.G Shin, Statistical Real-Time Channels on Multi-access Bus Networks, *IEEE Transaction on Parallel and Distributed Systems*, Vol.7(8), 769-780, 1997

[14] Z. Wang and T.R Wang, The Mechanism and Implement of FF Fieldbus DLL, AMSMA2000, 2000, 534-539

[15] Z. Wang, Modeling and Analysis of Fieldbus based Distributed Realtime System, PHD Dissertation 2000

Performance Investigation of ATM LANs

S. KAMOLPHIWONG
Centre for Network Research (CNR), Department of Computer Engineering,
Faculty of Engineering, Prince of Songkla University, Hatyai, Songkla, THAILAND 90112
Email: ksinchai@ratree.psu.ac.th
T. KAMOLPHIWONG
Centre for Network Research (CNR), Department of Computer Engineering,
Faculty of Engineering, Prince of Songkla University, Hatyai, Songkla, THAILAND 90112
Email: kthosapo@ratree.psu.ac.th
and
C. JANTARAPRIM
Linux Research Laboratory, Department of Computer Engineering,
Faculty of Engineering, Prince of Songkla University, Hatyai, Songkla, THAILAND 90112
Email: jchatcha@ratree.psu.ac.th

ABSTRACT

In this paper, we present a performance investigation of ATM LANs (Asynchronous Transfer Mode Local Area Networks). Three ATM LANs have been studied by using computer simulation: LANE (ATM LAN Emulation), CLIP (Classical IP over ATM), and MPOA (Multi-Protocol over ATM). The performance of each scheme is investigated in terms of time to deliver some amount of data where traffic load and end-to-end delay time conditions are varied. We have found that LANE gives the best result among all of them due to its simple mechanism, while CLIP performs not as good as MPOA since CLIP requires a number of information exchanges. However, when traffic load is high and round trip time is large CLIP and MPOA perform similar.

Keywords: ATM, LANE, CLIP, MPOA

1. INTRODUCTION

Recently, many network managers are investigating the benefits and challenges associated with migrating their networks to Asynchronous Transfer Mode (ATM). ATM's inherent capabilities such as gigabit-level bit rate, multiservice integration, virtual network support, and easy scalability make it an attractive alternative for network growth. However, network planners raise their caution flags when they consider how ATM technology will interoperate with their installed base of Ethernet and Token Ring equipment, data networking protocols, and legacy applications. This is a valid concern, since the ATM architecture differs fundamentally from legacy LAN technologies. ATM is a connection-oriented technology. It uses an abbreviated address, called a virtual channel identifier, to exchange data between two ATM stations over a virtual channel connection, or VCC. Legacy LANs, on the other hand, employ connectionless transmission technology based on global addressing. Most of today's data networking protocols have been designed to operate over such connectionless networks. Therefore, to use ATM for practical data networking, there must be some way of adapting existing network layer protocols, such as IP and IPX, to the connection-oriented paradigm of ATM.

Performance comparisons between various ATM LANs have been presented by a number of literatures. In [1], a comparison between ATM and DQDB was presented. A comparison of TM and Ethernet can be found in [3],[4], and between ATMLAN and FDDI [5]. Real experiment was setup to test the performance between LANE and MPOA [2]. However, some issues have not been pointed out, e.g. when end-to-end time is arid. Moreover, our results have shown a separated time of ach scheme internal process.

The rest of the paper is organized as follows: Next section, background information of ATM LANs are described. In section 3, simulation scenarios are presented. Simulation results are shown in section 4. We conclude our paper in section 5.

2 ATM LANs

• ATM LAN Emulator (LANE)

ATM LAN Emulation (LANE) is an ATM-based interne work technology that enables ATM-connected end stations to establish MAC-layer connections. LANE is a layer 2 bridging protocol that makes a connection- oriented ATM network look and behave like a shared connectionless Ethernet or Token Ring LAN segment. It allows existing LAN protocols, such as Novell NetWare, Microsoft Windows, DECnet, TCP/IP, MacTCP or AppleTalk, to operate over ATM networks without requiring modifications to the application itself.

The main objective of the LANE service is to allow existing applications to access the ATM network by way of MAC rivers as if they were running over traditional LAN's.

Standard interfaces for MAC device drivers include: NDIS, OI and DLPI. These interfaces specify how access to a AC driver is performed. Although the drivers may have different primitives and parameter sets, the services they provide are synonymous. LANE provides these interfaces and services to the upper layers.

Figure 1 shows where is a position of ATM in IEEE 802 standard.

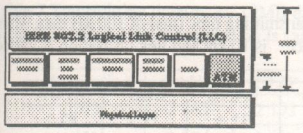


Figure 1 Overlay of ATM in IEEE 802.2

•Classical IP over ATM (CLIP)

RFC1577 [9] is basically designed to emulate IP sub-networks for stations connected on an ATM switch. RFC 1577 assumes a switched ATM environment (SVCs). In order to make the IP sub-network configuration easier, it defines ATMARP and inATMARP address resolution servers (One per IP subnet). Each station on an IP sub-network is configured with the ATM address of the ARP network server. When a station becomes active, it registers into the ARP server by placing a call to the server address. The basic encapsulation is compatible with RFC 1483: all messages carried on ATM are encapsulated with SNAP/LLC headers. RFC 1483 also defines how connectionless PDUs are transported over AAL5.The server obtains the IP address of the station by placing to it an in ATMARP call. It maintains a table associating IP and ATM addresses. RFC 1755 [12] explains the ATM signaling involved in IP over ATM operation.

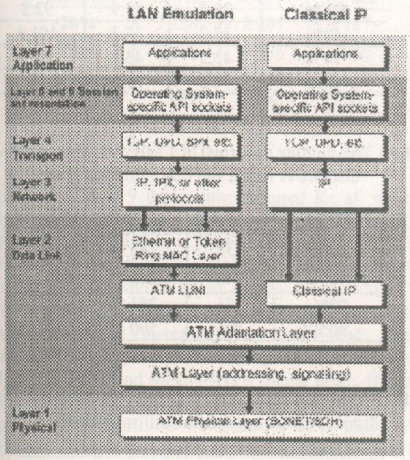


Figure 2 comparison of LANE and Classical IP

• Multi-protocol over ATM (MPOA)

Classical IP over ATM offers a solution for the emulation of IP sub networks over ATM. An extension is needed, to allow the emulation of any protocol over ATM. The ATM Forum has established a working group to consider the development of multiprotocol over ATM (MPOA) standards [11]. Figure 3 shows MPOA's components This will provide a means for extending native-mode protocol support beyond IP. This is the purpose of the MPOA task force in the ATM forum. In fact, MPOA and I-PNNI pursue the same objectives through different techniques, and it is not excluded that other variants may be proposed. The objectives of the MPOA task force are the following:

- Allow the operation of all LAN protocols over ATM, and not only of TCP/IP,
- Allow the interworking of legacy-LAN attached stations, and of ATM attached stations,
- Allow protocols and applications to use the new characteristics provided by ATM, like Quality Of Service management,
- Be usable on very large networks (Scalability).

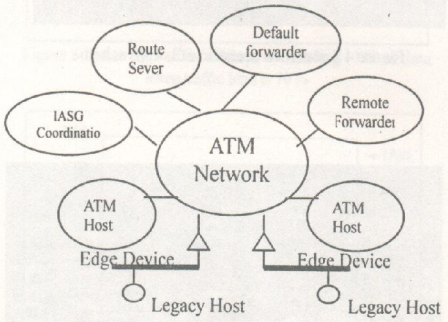


Figure 3 MPOA's components

The raise of paper is organized as follows: In next section, computer simulation models of all three schemes will be presented. In section 3, simulation results will be given, A discussion of simulation results is given in section 4. We conclude our work in section 5.

3. Simulation Models

In this session, simulation models for LANE, CLIP, and MPOA are presented. A computer simulation tool [6] is used for this purpose. The following parameters are used for all simulation scenarios:

- All Ethernet packets are encapsulated by AAL5,
- Traffic policing in ATM is based on UNI 3.0,
- Each ATM link bandwidth is 155 Mbps,
- Average Ethernet packet size is 1.5 kB,
- A number of source and destination pairs is 5,
- Each Ethernet source generates traffic load up to 30 Mbps,
- All simulation time is 20 sec,
- End-to-end distance is varied, 10, 100, and 1000 km,
- Traffic load is varied, 20, 70 and 90 percent,
- Block of transmitted data is varied, 100kB, 1MB, and 10 MB.

Figure 4 shows a simulation scenario of LANE scheme. We can see that LANE's components are constructed and attached to ATM server. Figure 5 depicts a simulation scenario of CLIP. The most important component is ARP unit. Figure 6 shows a simulation scenario of MPOA. We can notice that its complexity is higher two previous schemes.

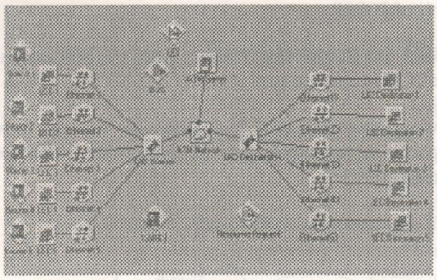


Figure 4 Simulation scenario of LANE scheme

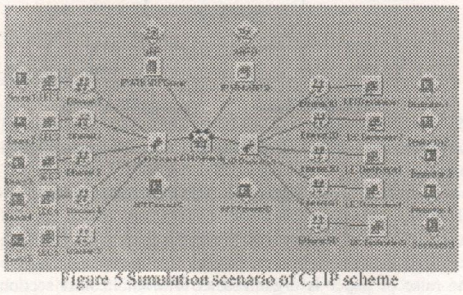


Figure 5 Simulation scenario of CLIP scheme

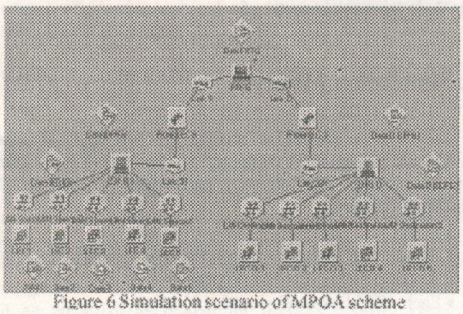


Figure 6 Simulation scenario of MPOA scheme

2. SIMULATION RESULTS

In this section, we present the simulation results obtained from simulation scenarios described in 3.

Table 1 to Table 3 show data transmitting time required for each scheme for transmitting 100 kB of data when end-to-end distance is varied from 10 km to 1,000 km. LANE and CLIP give a similar result while MPOA gives almost 3 times of required time higher than LANE and CLIP.

In Table 4 to Table 6, 1 MB of data are transmitted and distance is varied from 10 km to 1 000 km. LANE and CLIP give a similar result while MPOA requires 3 times larger than the one used by LANE and CLIP. Again, when a size of data is as large as 10 MB, the similar results are obtained as shown in Table 7, Table 8, and Table 9.

Table 1 Transmitting time (ms) of 100 kB data when end –to-end distance is 10 km and link utilisation is varied.

Utilisation	LANE	CLIP	MPOA
20%	31.4	31.2	87.2
40%	32.7	32.4	96.0
70%	37.7	37.4	100.3
90%	66.6	66.1	200.7

Table 2 Transmitting time(ms) of 100 kB data when end-to-enddistance is 100kmand link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	31.9	31.2	87.7
40%	33.2	32.5	96.5
70%	38.3	37.5	100.8
90%	67.6	66.2	201.7

Table 3 Transmitting time(ms)of 100 kB data when end-to-End distance is 1,000 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20.%	36.2	35.9	92.2
40%	37.7	37.3	101.4
70%	43.5	43.0	106.0
90%	76.8	76.0	212.0

Talbe 4 Transmitting time (ms) of 1 MB data when end-to-end distance is 10 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	238	236	858
40%	248	245	944
70%	286	283	987
90%	505	500	1,974

Table 5 Transmitting time(ms) of 1MB data when end-to-end distance is 100 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	242	240	858
40%	252	249	944
70%	291	287	987
90%	513	508	1 974

Table 6 Transmitting time(ms)of 1MB data when end-to-end distance is 1,000 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	246	245	863
40%	256	255	949
70%	296	294	992
90%	522	520	1 985

Table 7 Transmitting time(ms)of 10MB data when end-to-end distance is 10 km and link utilization is varied

Utilisation	LANE	CLIP	MPOA
20%	2 283	2 193	8 182
40%	2 374	2 281	9 000
70%	2 740	2 632	9 409
90%	4 840	4 649	18 819

Table 8 Transmitting time(ms)of 10MB data when end-to-end distance is 100 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	2 284	2 198	8 186
40%	2 375	2 386	9 004
70%	2 710	2 638	9 419
90%	4 841	4 660	18 827

Table 9 Transimitting time(ms)of 10MB data when end-to-end distance is 1,000 km and link utilization is varied.

Utilisation	LANE	CLIP	MPOA
20%	2 288	2 207	8 187
40%	2 380	2 296	9 006
70%	2 746	2 649	9 415
90%	4 851	4 680	18 830

Figure 7 to Figure 9 depict time required for all steps for sending 1 MB of data when traffic load is varied from 20, 70, and 90 percent respectively. We can see that for a light load traffic condition, e.g. 20 percent as shown in Figure 7, LANE performs best, and MPOA is just slightly lower. While CLIP performs almost 50 percent worst than the formers. The time required for CLIP increases when end-to-end distance increases. When traffic is moderate, e.g. 70 percent as shown in Figure 8, LANE and MPOA give a similar result. However, when traffic load condition is high, e.g. 90 percent as shown in Figure 9, MPOA performs worst among all of them. When end-to-end distance is 1,000 km required time for CLIP is higher than MPOA.

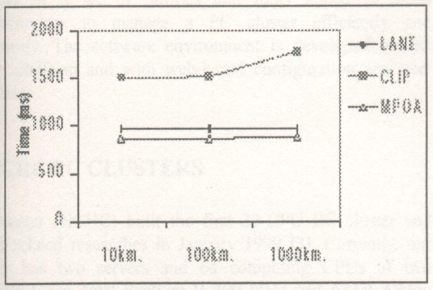


Figure 7 Time required for all steps for sending 1 MB of data when traffic load is 20%

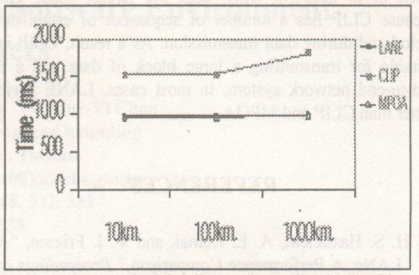


Figure 8 Time required for all steps for sending 1 MB of data when traffic load is 70 %

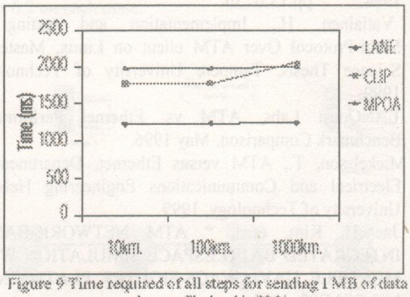


Figure 9 Time required of all steps for sending 1 MB of data when traffic load is 90 %

5. RESLUT DISCUSSION

In the simulation results, we have found that LANE and CLIP give a similar data transmitting time regardless of size of data. This is because LANE and CLIP have a similar architecture. Time consumed during inter-processes is less significant when compared with end-to-end delay time, and this seem to be independently with size of data. In contrast, MPOA has more sequences than the formers during channel establishment period.

The simulation results show that CLIP takes a longer time than LANE and MPOA . This figure is much larger when end-to-end delay time increases. This is because CLIP employs ARP (Address Resolution Protocol) where contacting from/to a server is frequently. While MPOA has a shorter communication than CLIP, as a result, MPOA gives a better overall results. However, when traffic intensity and end-to-end delay increase CLIP and MPOA perform similar. LANE performs best among all of them duce to its low complexity.

6. CONCLUSION

In this paper, we present a performance investigation of ATM LANs using computer simulation. Three ATM LAN technologies are examined: LANE (ATM LAN Emulator), CLIP (Classical IP over ATM), and MPOA (MultiProtocol over ATM). The performance of each scheme is considered in the following conditions: size of data, end-to-end delay time, and traffic load intensity. We have found that MPOA takes the longest time during data transmission period, excluding channel setup time, from one end to the other end. While CLIP takes the shortest time of data transmission. However, when the whole process (including channel establishment period) is

taken in to account, CLIP uses the largest time. This is because CLIP has a number of sequences of establishment period and during data transmission. As a result, CLIP is not suitable for transmitting a large block of data and a large end-to-end network system. In most cases, LANE performs better than CLIP and MPOA.

REFERENCES

[1] H. S. Hassanein, A. E. Kamal, and V. J. Friesen, "ATM LANs: A Performance Comparison," *Proceedings of the Third International Conference on Computer Communications and Networks*, San Francisco, Sept. 1994, pp.133-139

[2] Vatiainen, H., Implementation and testing of Multi-Protocol Over ATM client on Linux, Master of Science Thesis, Tampere University of Technology, 1999.

[3] LANQuest Labs, ATM vs. Ethernet Performance Benchmark Comparison, May 1996.

[4] Mickelsson, T., ATM versus Ethernet, Department of Electrical and Communications Engineering Helsinki University of Technology, 1999

[5] Jae H. Kim, et.al, " ATM NETWORK-BASED INTEGRATED BATTLESPACE SIMULATION WITH MULTIPLE UAV-AWACS-FIGHTER PLATFORMS," MILCOM'98, 1998

[6] CACI Product Company, "COMNET III, Release 2.0," 1998.

[7] Fore System Co. Ltd., "LAN Emulation, Virtual LANs, and the ATM Internetwork, Version 1.0,"

[8] ATM Forum, "LAN Emulation-Over-ATM v1.0 Specification," 1997.

[9] RFC 1577, Classical IP and ARP over ATM, IETF, 1994

[10] C. Brown, "Baseline text for MPOA," ATMF/95-0824R6, February 26, 1996.

[11] ATM Forum Technical Committee. *Multi-Protocol Over ATM - Version 1.0.* July 1997.

[12] M. Perez, F. Liaw, A. Mankin, E. Hoffman, D. Grossman, A. Malis, ``ATM Signaling Support for IP over ATM", RFC 1755, IETF February 19

A Monitoring and Management Software Environment for PC Clusters

Hsi-Ya Chang Kuo-Chan Huang Chaur-Yi Chou
National Center for High-Performance Computing
P.O. Box 19-136, Hsinchu, Taiwan
Email:{ c00jhc00,c00kch00,b00cyc00}@nchc.gov.tw
Tel: +886-3-5776085 Ext. 368, 312, 332
Fax: +886-3-5773538

ABSTRACT

PC cluster has recently emerged as a cost-effective parallel computing platform. As PC cluster gets popular, the management work on it has become an important issue. This paper presents a cluster monitoring and management software environment, which consists of a set of utilities and tools aimed to assist administrators to manage a PC cluster efficiently and effectively. The software environment is developed on the Linux platform and with web-based configuration and user interfaces.

Keywords: PC cluster, Linux, Web-based User Interface, Monitoring and Management Software

1. INTRODUCTION

In recent years the constant improvement of performance and quality of commodity PC hardware and networking devices has made PC clusters [1, 2] emerging as one of the most attractive computing platforms for many scientific and engineering applications. In the beginning of 1999 we set up our first generation PC cluster with 32 CPUs at the National Center for High-Performance Computing (NCHC) in Taiwan, and started several plans for research and development of related enabling technologies.

With its low-cost advantage, PC cluster has become a viable and popular solution for the ever-increasing demand of computing power in small organizations and laboratories. In Taiwan, many organizations and laboratories have built or are going to build their own PC clusters to fulfill the computation need. As PC cluster gets popular, the management work on it has become an important issue. This paper presents a cluster monitoring and management software environment, which consists of a set of utilities and tools aimed to assist administrators to manage a PC cluster efficiently and effectively. The software environment is developed on the Linux platform and with web-based configuration and user interfaces.

2. NCHC PC CLUSTERS

Our center (NCHC) built the first 32-CPU PC cluster and began related researches in January 1999 [3]. Currently, our cluster has two servers and 64 computing CPUs of two different types, Intel Pentium II 400 MHz and AMD Athlon 750 MHz. The interconnection network inside the cluster is 100 Mbps Fast Ethernet. The network topology of our PC cluster is illustrated in Figure 1, which consists of four stackable 24-port switching hubs. The cluster runs redhat Linux 6.2 on each node.

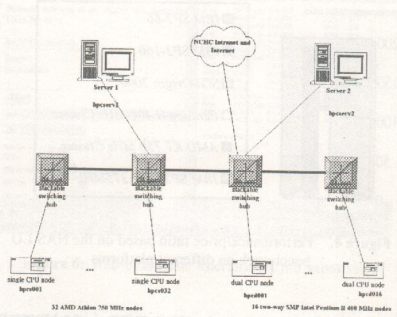


Figure 1. Network topology of NCHC PC cluster

Figure 2 shows the performance of the NAS parallel benchmark [4]. The figure indicates that the performance of the PC cluster can scale well up to at least 32 nodes for commonly used application programs. Figure 3 and 4 together show that PC clusters deliver performance comparable to traditional parallel computers and have the best performance/price ratio. The figures also indicate that the performance of PC clusters increases quickly every year because of the continuous improvement of the commodity components of PCs and networks. The above data illustrate that PC clusters are comparatively cost-effective; this explains why PC clusters are becoming a popular alternative platform for parallel computing.

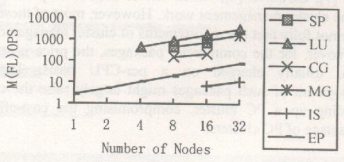


Figure 2. Performance of NAS parallel benchmark on NCHC PC cluster

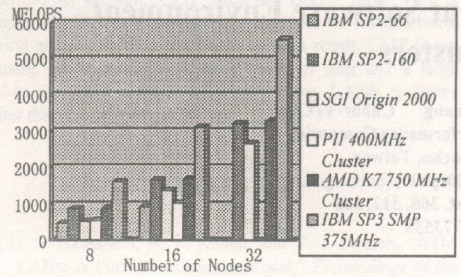


Figure 3. Parallel performance of the NAS LU benchmark on different platforms

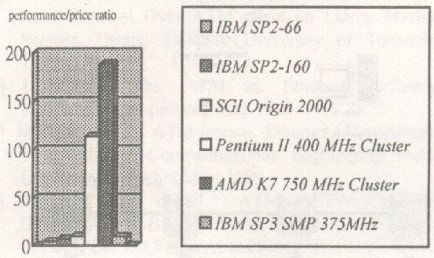


Figure 4. Performance/price ratio based on the NAS LU benchmark on different platforms

3. REQUIREMENT OF THE CLUSTER MONITORING AND MANAGEMENT SOFTWARE ENVIRONMENT

Most current PC clusters run an independent copy of the operating system, such as Windows NT or Linux, on each node. To coordinate the operation of the whole cluster, it is therefore necessary to employ another layer of management software on top of all the running operating system instances. It is this layer of software that transforms a set of connected computers into a cluster environment. This paper focuses on the type of cluster which aims to provide a parallel computing environment for running batch parallel jobs. This is the common usage pattern for scientific and engineering researchers conducting computational simulation or problem solving.

Currently, there are some public-domain software tools such as DQS [5], PBS [6], and other commercial packages, e.g., LSF [7], CODINE [8], which can be used to help some parts of the cluster management work. However, most of these tools have not fully met the requirements of cluster management yet. Moreover, for the commercial packages, the price is still too high, usually charged on a per-CPU basis. Therefore deployment of such packages might largely raise the cost of building up a PC cluster, compromising the cost-effective advantage of PC clusters.

Before developing our own software environment, we had been using DQS [5] as a supporting tool for cluster management. DQS provides functionalities for job scheduling and queue management, but in many aspects, such as reliability and access control, there is still much room for improvement. Based on the experience in operating the NCHC PC clusters, we decided to build our own cost-effective cluster management software environment, which is aimed to provide a convenient environment for users to run computation jobs and for system administrators to monitor and manage the cluster. The following is a list of requirements that we consider important in a cluster management software environment.

- *User-friendly interface*. A graphical user interface is useful for administrators and users to easily understand and operate the system utilities.
- *Portable monitoring and management*. It's convenient for administrators to be able to monitor and manage the clusters from any platform at anytime anywhere. Providing a web-based interface is an effective first step toward this goal.
- *Event notification facilities*. Either by e-mails or pagers, the notification facilities can release administrators from keeping watching the management console all day long for noticeable system information, and still allow them to effectively manage any emergent situation.
- *Automatic handling*. Some usual and routine system configuration activities in response to certain events can be automated to improve the management efficiency and leave the administrators for solving more complex problems.
- *Access control*. It is an important issue how to detect or prevent users from bypassing the cluster management system to directly execute jobs on certain computing nodes.
- *Efficient and fair job scheduling*. Advanced scheduling algorithms are needed to increase system utilization and avoid unfair starvation situations especially for parallel jobs.
- *Node crash handling*. If a node crashes while some job is running on it. The management software is responsible for properly handling the crash situation, restarting the interrupted job on another node and rebooting the crashed node for necessary maintenance.

4. SYSTEM ARCHITECTURE

This section presents the functionality and architecture of the cluster monitoring and management software environment developed at our center. The first version of this software has been released to interested users in Taiwan, while further improvement and enhancement are continuously under development. The software environment has a layered architecture as shown in Figure 5.

The software environment is divided into four collaborative subsystems: scheduling, logging, monitoring, and configuration subsystems. The scheduling subsystem is responsible for accepting user's job submission and arranging the job execution properly to achieve maximum system utilization and user satisfaction. For parallel jobs, the scheduling subsystem currently supports parallel programs written with PVM [9] or MPI [10]. The scheduling subsystem provides flexible mechanisms for priority arrangement of both jobs and users. The administrator can assign static priorities for users at the system configuration time, while at runtime users or administrators can dynamically tune the execution order of a set of jobs according to their granted privileges. Suspend-and-restart mechanism is deployed to implement

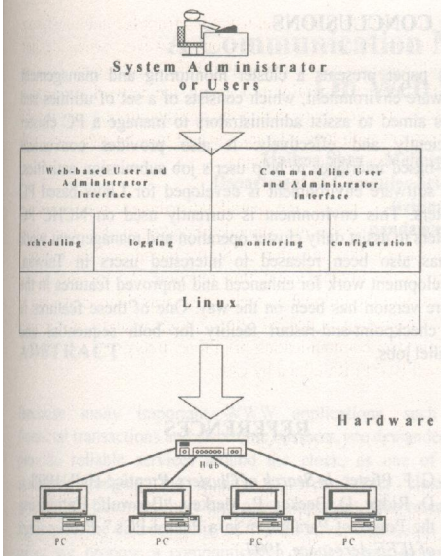


Figure 5: Layered architecture of the cluster monitoring and management software environment

advanced scheduling policies for avoiding job starvation situations as well as maintaining high system utilization. The scheduling subsystem will automatically perform a job resubmission once a job's execution is interrupted by node crashes or other accidents. More advanced checkpoint-and-restart mechanisms are under development to enhance system reliability and save job re-run time.

The logging subsystem keeps every job submission, execution, completion, and deletion record during the system operation. The subsystem also provides tools for analyzing these log data to discover useful information, which can help adjust the system configuration for better performance and utilization.

The monitoring subsystem keeps monitoring resource usage, resource availability, and process status on each computing node. Administrators can view the whole cluster status through a web-based interface. An automatic event notification mechanism is deployed so that once a predefined noticeable situation occurs; administrators will get an immediate message through e-mail or paging. Administrators can also define a handling routine for each specific event; the monitoring subsystem will automatically trigger this handling routine for instant trouble shooting when the corresponding event occurs.

The configuration subsystem includes a set of web-based interfaces. If needed, the configuration subsystem can help to divide a cluster into a set of sub-clusters for different purposes, and help to manage the set of sub-clusters smoothly. Administrators can easily adjust system configurations, from cluster member setting, access control, to user and job priorities, through these web pages. Each time administrators change some part of the system configuration, the subsystem will automatically notify the corresponding running daemons in the cluster to update their configuration information immediately for proper system operation.

5. CURRENT ENVIRONMENT

This section presents some snapshots of current cluster monitoring and management software environment. Figure 6 is the web-based user interface for job submission. In the right frame, users can view the cluster status, such as job running or queuing information and available CPUs. In the left frame, users can specify the properties of the jobs they want to run, submit the jobs, or delete a running job. The user interface is developed using HTML web pages together with a set of CGI programs, which cooperate with the corresponding daemons on the server node.

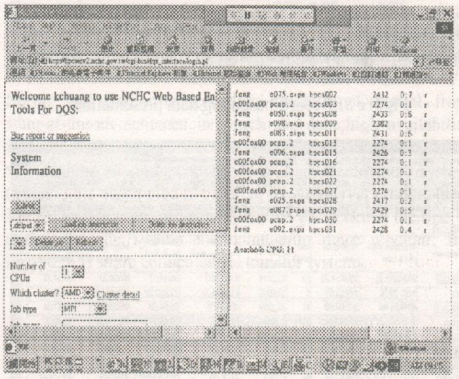


Figure 6: Web-based user interface for job submission

Figure 7 is the interface for log data query. Administrators can query log data with specified conditions or perform simple statistical analysis on the log data through the interface. Figure 8 is a record-based log data presentation. Through this presentation, administrators can thoroughly investigate each job record to look for abnormal usage patterns or perform detailed analysis. Administrators can also perform statistical analysis on the log data to make some meaningful observations, such as who consumes the most system resource, averaged execution time, waiting time, and parallelism, ...etc.. Figure 9 is an example of such kind of analysis.

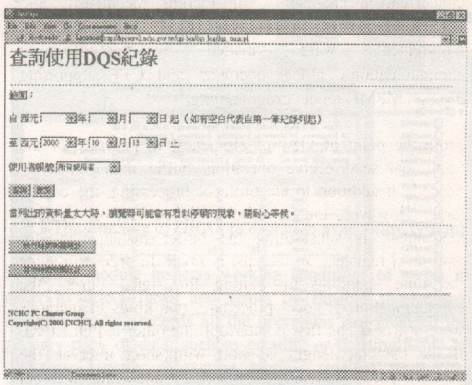


Figure 7: Web-based interface for log query

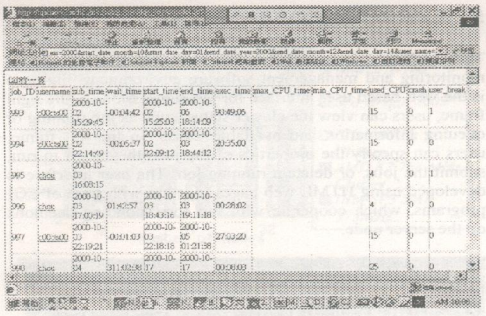


Figure 8: Record-based log data presentation

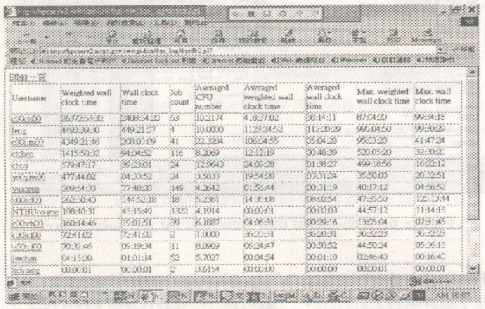


Figure 9: Statistical log data-analysis

Figure 10 is an example of the monitoring interface. Through this interface, administrators can check the status of the whole cluster, and perform some maintenance activities on specific nodes in the cluster. The interface is implemented as a Java applet which can be downloaded to a Java-enabled web browser for execution. The applet communicates with various serving processes in the server to provide collected cluster status and execute maintenance commands specified by administrators.

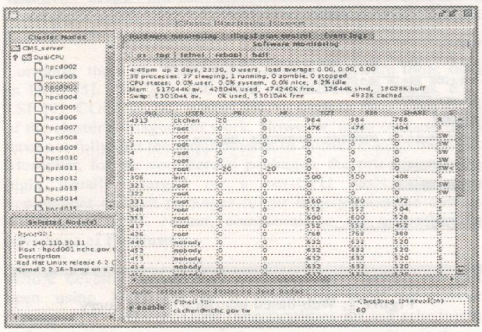


Figure 10: An example of monitoring interface

6. CONCLUSIONS

This paper presents a cluster monitoring and management software environment, which consists of a set of utilities and tools aimed to assist administrators to manage a PC cluster efficiently and effectively. It also provides convenient web-based interfaces to help user's job submission activities. The software environment is developed for Linux-based PC clusters. This environment is currently used on NCHC PC clusters to assist daily cluster operation and management work. It has also been released to interested users in Taiwan. Development work for enhanced and improved features in the future version has been on the way. One of these features is the checkpoint-and-restart facility for both sequential and parallel jobs.

REFERENCES

[1] G. F. Pfister, *In Search of Clusters*, Prentice-Hall, 1998.

[2] D. Ridge, D. Becker, P. Merkey, "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs." *Proceedings of IEEE Aerospace*, 1997.

[3] K. C. Huang; H. Y. Chang; C. Y. Shen; C. Y. Chou; S. C. Tcheng, "Benchmarking and Performance Evaluation of NCHC PC Cluster," APSCC'2000 with HPCAsia2000, May 14-17, Beijing, China.

[4] http://science.nas.nasa.gov/Software/NPB

[5] http://www.scri.fsu.edu/~pasko/dqs.html

[6] http://pbs.mrj.com/

[7] http://www.platform.com

[8] http://www.gridware.com

[9] A. Geist, A. Beguelin; J. Dongarra *et. al.*, *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, 1994.

[10] DSW. Gropp, E. Lusk; A. Skjellum, Using MPI: Portable Parallel Programming with Message-Passing Interface, The MIT Press, 1994

A Communication Model for Data Availability On Web Server Clusters*

Haihua Shen Meiming Shen Weimin Zheng Shuming Shi
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, P. R. China
panda@est4.cs.tsinghua.edu.cn

ABSTRACT

Because many important WWW applications, such as financial transactions and electronic business, are demanded to provide reliable services around the clock, as one of the standards judging the quality of Web server clusters, availability becomes more and more important. Our research focuses on the data reliability of WWW server clusters. In this paper we propose a communication model using automata theory for Data Availability on Web Server Clusters and apply it in DHAS (a data high-availability system we developed). Our results indicate that the system keeps good scalability as well as availability and the cost can be decrease to a tolerable level by using the communication model.

Keywords: Web Server Clusters, High Availability, Cluster File System (Clufs), Modular, Nondeterministic, Finite Automata.

1. INTRODUCTION

The fast development of Internet demands more and more powerful servers. Comparing with the expensive high performance host or SMP, Web server clusters have more scalability, less cost and satisfying response speed. So many famous Web sites have thrown away single servers and turned into Web server clusters. Basically web server clusters are composed of a set of processing nodes, each with a local disk array, connected by a high bandwidth switch or network. (See. .Fig1).

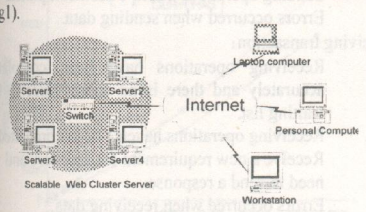


Fig 1. The basic structure of Web server clusters

Ideal Web server clusters should have many important characteristics: high performance, good scalability, low cost, availability and manageability [1]. Because many important WWW applications, such as financial transactions and electronic business, are demanded to provide reliable services around the clock, as one of the standards judging the quality of Web server clusters, availability becomes more and more important [2]. It demands that when some machine nodes in a cluster fails, other nodes can provide services as more as possible. A good highly available scheme for a cluster web server involves many aspects: the reliability of Request-dispatcher, WWW server and network, the reliability of software and data. And our research focuses on the WWW Server and first considers data reliability. Different from state-of-the-art schemes in which the data high availability implementation is embedded in the file system itself [3][4][5], we design an independent data high-availability system (DHAS) for distributed file system on web server clusters (see. Fig 2). Two benefits are achieved from this scheme: the first, it makes the file system implementation more modular; the second, it is more convenient to transfer systems.

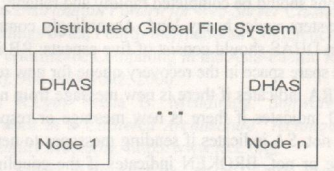


Fig. 2 The independent data high-availability system (DHAS) for distributed file system

DHAS is a middle layer between the clusters' operating system and distributed global file system and uses data replication to provide data availability while still maintaining high transaction throughput using write through. The system consists of fore important parts: snooping module, logging module, communicating module and recovering module (see Fig.3).

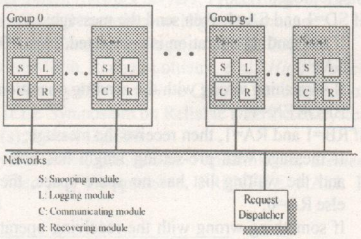


Fig.3 The structure of DHAS (a data high-availability system)

Snooping module inspects working conditions of nodes in order to detect flaws and errors by timeout mechanism. Logging module records all the operations that changed the primary copy of data, such as creating or deleting a file or directory. Usually recovering module receives recovering information from the communicating module and recovers replicas on the local node according to the primary data on coupling nodes. When something wrong with the system, recovering module can keep data synchronous and help system

reconstructing. Communicating module provides rapid and reliable information transfer and ensured the on-line data replication mechanism. In this paper we focus on the problem of maintaining good transaction throughput in spite of having to update the backup. Therefore it is necessary for us to study communication models for data availability on web server clusters.

The rest of the paper is organized as follows. Section 2 makes theoretical analysis on status message in course of communication. Based on these analyses, we put forward a model suitable to the communication of DHAS in terms of automata theory in section 3. Then we conclude in section 4 with some general observations on the importance of our research.

2. ANALYSIS OF COMMUNICATION STATUS IN DHAS

Communication operations in DHAS include getting original recovery message, sending recovery message to the network, receiving recovery message from the network, putting the recovery message received into the recovery queue and carrying out the instruction from snooping modules. All these operations should be completed rapidly and reliably, especially when system under heavy load. Accordingly communication status in DHAS should consist of five aspects: RB indicates if there is spare space in the recovery queue for new requirement or not, RA indicates if there is new message from networks or not, SD indicates if there is new message or response to be sent or not, SA indicates if sending messages to networks can be done or not, BROKEN indicates if the coupling node is available or not.

The following algorithm provides the description for communication operations in DHAS.

Begin of the loop;
If there is new message, then SD=1;
If an operation in recovery queue has been done, then release the possessive space and RB=1;
If BROKEN=1, then the system will be turned into working with single-node;
Else if SD=1 and SA=1, then send the message;
    If the sending operation is completed, then SD=0;else SA=0;
    If something wrong with the sending operation, then BROKEN=1;
Else if RB=1 and RA=1, then receive the message;
    If message with pre-setting length has been received and the waiting list has no spare space, then RB=0; else RA=0;
    If something wrong with the receiving operation, then BROKEN=1;
Else if RB=0 and RA=1, then wait for recovery requirement being done;
Else if RA=0 and SA=0, then send (or receive) message to (or from) networks or virtual devices synchronously (i.e. the process is blocked until get response in terms of timeout mechanism) and set SA,RA,BROKEN according to the value returned;
Else if RA=0, SD=0 and SA=1, then block the operation of sending (or receiving) message to (or from) networks or virtual devices by select( ) and set RA according to the value returned;
End of the loop;

By analyzing the algorithm above, we can easily find the corresponding relation between status and operations of communication and the state diagram of an automaton. So it is naturally to study the communication mechanism using the theory of automata [7].

3. DESIGN OF COMMUNICATION MODEL IN DHAS USING AUTOMATA THEORY

According to the analysis in section 2, we present a nondeterministic finite automaton M=(K, Σ, δ, q0, F), where

K is a finite set of states,
Σ is an alphabet,
δ, the transaction function, is a function from KXΣ to K,
q0∈K is the initial state, and
F⊆K is the set of final states.

The states of the automaton are represented by <RB, RA, SD, SA, BROKEN>, where RB=1 indicates that there is spare space in the recovery queue for new requirement, RA=1 indicates that there is new message from networks, SD=1 indicates that there is new message or response to be sent, SA=1 indicates that sending messages to networks can be done, BROKEN=1 indicates the coupling node is not available according to timeout mechanism. Then according to communication status analyzed in section 2, K={q0,q1,q2,q3,q4,q5,q6,q7,q8,q9 } and F={q9}, where

q0:	<1,0,0,1,0>	q5:	<1,1,1,1,0>
q1:	<1,0,1,1,0>	q6:	<0,1,1,1,0>
q2:	<1,0,1,0,0>	q7:	<0,1,1,0,0>
q3:	<0,1,0,1,0>	q8:	<1,1,1,0,0>
q4:	<1,1,0,1,0>	q9:	<x,x,x,x,1>

Σ={0,1,2,3,4,5,6,7,8,9,10,11}, where

0 There is new recovery message which coming
1 There is new recovery message to be sent.
2 There are operations that have been done in the recovery queue.

Sending transaction:

3 Sending operations have been completed accurately.
4 Sending operations haven't been completed.
5 Errors occurred when sending data.

Receiving transaction:

6 Receiving operations have been completed accurately and there is no spare space in the waiting list.
7 Receiving operations haven't been completed.
8 Receive a new requirement for recovery and need to send a response.
9 Errors occurred when receiving data.
10 Block the operation of sending (or receiving) message to (or from) networks or virtual devices by select( ) in terms of timeout mechanism until networks are available.
11 Block the operation of sending (or receiving) message to (or from) networks or virtual devices by select( ) and set RA according to the value returned.

Accordingly the communication model can be depicted by means of state diagrams (see Fig.4). We haven't list all inputs in Σ for each state q of M because many communication

operations are only relevant to a few states, namely, those inputs that can not cause state transitions are ignored. The robustness of the model has been improved naturally since exception handles extracted from the practical algorithm have been imbedded in the model beforehand.

Using the communication model, we develop a data high-availability system (DHAS), which can successfully support a distributed file system we developed, named Clufs (Cluster File System).

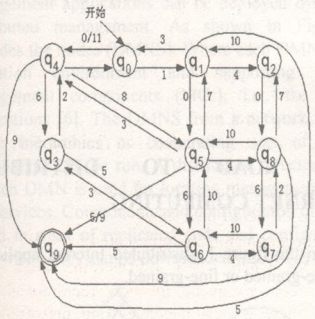


Fig.4. The state diagram of the communication model for data availability on web server clusters

4. CONSLUSIONS

We propose a communication model for Data Availability on Web Server Clusters and apply it in DHAS (a data high-availability system we developed). At present, it is popular to implement the data high availability in the file system. But we design an independent data high-availability system (DHAS) for two reasons: the first, it makes the file system implementation more modular; the second, it is more convenient to transfer systems. Though making a high-availability system independent may brings additional cost on communication, our results show that the cost can be

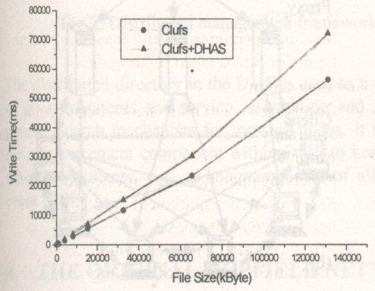


Fig.5 Comparing write time performance of the system with DHAS using the communication model with that of the original system

decrease to a tolerable degree by using the communication model (see Fig.5.). We also provide the resulting performance measures to illustrate that the system keeps good scalability as well as availability by using the model (see.Table.1).

Table 1: The write time performance of DHAS for a system with 2,4 or 8 nodes

Size	2 nodes	4 nodes	8 nodes
	ms	ms	ms
1k	16	17	17
4k	17	18	18
16k	20	21	22
64k	41	43	46
256k	126	130	134
512k	237	241	246
1M	464	469	472
4M	1852	1858	1862
8M	3580	3584	3588
16M	6971	6977	6980
32M	15189	15220	15232
64M	30175	30196	30216
128M	72029	72063	72102

The table above indicates that the system using the communication model has good scalability.

REFERENCES

[1] Shen Haihua; Chen Shimin, et al., *Research on Data Replica Distribution Pattern for Web Server Clusters*, Proceedings of the Fourth International Conference on High Performance Computing in the Asia-Pacific Region, Page(s): 966 -968 vol.2, 2000.

[2]Michael R.Lyu; Veena B. Mendiratta. *Software Fault Tolerance in a Clustered Architecture: Techniques and Reliability Modeling*, http://www.ieee.org.

[3]Cristiana Amza, Alan L. Cox, et al., *Data Replication Strategies for Fault Tolerance and Availability on Commodity Clusters*, Proceedings of International Conference on Dependable Systems and Networks, Page(s): 459 -467, 2000

[4] Chu-Sing Yang, Mon-Yen Luo. *Building an Adaptable, Fault Tolerant, and Highly Manageable Web Server on Clusters of Non-dedicated Workstations*, Proceedings of International Conference on Parallel Processing, Page(s): 413 -420, 2000

[5] Enrique V.Carrera, Ricardo Bianchini, *Evaluation Cluster-Based Network Servers*, Proceedings of The Ninth International Symposium on High-Performance Distributed Computing, Page(s): 63 -70, 2000

[6] Christine Morin, Renaud Lottiaux, et al., *High Availability of the Memory Hierarchy in a Cluster*, Proceedings of the 19th IEEE Symposium on Reliable Distributed Systems, Page(s): 134 –143,2000

[7] Harry R. Lewis, Christos H. Papadimitriou, *Elements of The Theory of Computation*, PRENTICE HALL Press, 1998

* The work described in this paper is supported by National Natural Science Fund (PR. China) 60073010.

Active Networks for Efficient and Distributed Network Management Based on an Artificial Neural Network*

Youwei Yuan La-mei Yan
Department of Computer Science & Technology, Zhuzhou Engineering Institute
Hunan, 412008, China E-mail: yuanyouwei@163.net

ABSTRACT

The distributed management framework (DMF) presented in this paper provides an environment that allows a broad range of management tasks to move and run anywhere within the managed system. The paper presents a new approach based on the Hopfield model of artificial neural networks to solve the routing problem in the distributed computer networks design. The proposed method can find the best path between any node pair by minimizing an energy function. Comparison with the other optimal routing algorithms, the mean delays obtained by the neural-network approach are generally more stable and considerably smaller in execution. As a result, the proposed neural-network approach is suitable to be integrated into overall topological design processes, for moderate and high-speed networks subject to quality of services constraints as well as to changes in configuration and link costs.

Keywords: Distributed Management; Active Network Management; Neural Network; Routing Algorithm. Mobile Agent

1. INTRODUCTION

Our world is becoming increasingly heterogeneous, decentralized and distributed, but the software that is supposed to work in the world, usually ,is not .The business world is the center of "disparate computing". [3]Distributed software systems are often more complicated and less portable than single-user applications. The distributed management framework (DMF) presented in this paper provides an environment that allows a broad range of management tasks to move and run anywhere within the managed system. The nodes through which the packet is transmitted from the source to the destination constitute a path or a route, and the mechanism used to select one route among various alternatives to link each source-destination pair is called a routing procedure.[4] The proposed method based on Hopfield artificial neural networks can find the best path between any node pair by minimizing an energy configuration. Comparison with the other optimal routing algorithms, the mean delays obtained by the neural-network approach are generally more stable and considerably smaller in execution.[4]

This article is structured as follows: it begins with an overview of DMF and the routing problem. Section 2 presents the road to distributed Internet computing. Section 3 presents the distributed managements framework. Section 4 presents the original Hopfield neural network. Section 5 presents routing in distributed computer networks by Hopfield neural networks. Section 6 exposes the implementation and comparison with optimal routing algorithms.

2. THE ROAD TO DISTRIBUTED INTERNET COMPUTING

As shown in the figure 1, distributed Internet applications can be coarse-grained or fine-grained.

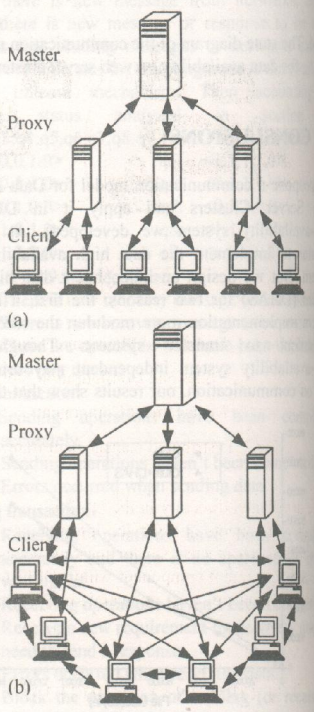


Fig.1. The constitute of distributed

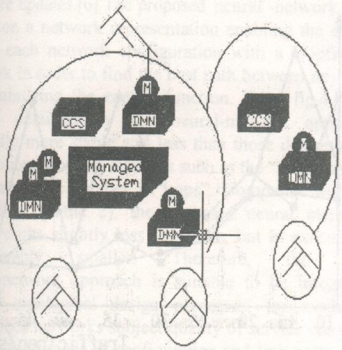
In coarse-grained applications, clients communicate with each other. This approach is good for brute-force computations in which the application only has to divide calculations among different clients.

In fine-grained applications, participating clients must communicate with each other. This is necessary for such applications as simulations. As simulations progress, changes in situations handled by other. Clients thus must be able to communicate these changes to each other.

3. A FRAMEWORK FOR DISTRIBUTED MANAGEMENT

The alternative to centralized management is the distribution of management tasks across the management system to reduce bottlenecks. [1]

The distributed management framework (DMF) provides an object-oriented architecture where lightweight. Mobile management applications can be deployed dynamically for distributed management. As shown in Fig.2.the DMF provides the nodes (DMNS). Where each DMN provides the execution environment and supporting services for management components (MCs), i.e.. the management applications [6]. The DMNS from a network of peers over which hierarchies or cooperating sets of management components can be run. A distributed directory maintained by each DMN is used for locating management components and services. Components and configuration of the DMF are stored in a set of replicated code and configuration stores, from where they can be downloaded to DMNS as required.



DMN: Distributed Management Nodes

CCS: Codes and Configuration Store

M: Management Component

Fig.2. Distributed management framework

The distributed directory in the DMF is used as a depository of all components and service information and supports a generic search mechanism for locating nodes. It is possible for a management component within DMF to keep track of other components. e.g., to maintain a list of all available event filters.

4. THE ORIGINAL HOPFIELD NETWORK

A Hopfield network consists of n completely connected neurons. Each neuron has two possible states: V_i=-1 and V_j=1.The connection of neuron i to neuron j is denoted by T_{ij} and the total entry of a neuron i is equal to $\sum_j T_{ij} V_j$.

The energy of a Hopfield and Tank (1986) network is defined as:

$$E = -\frac{1}{2} S,$$

Where

$$S = V_1(w_{12}V_2 + w_{13}V_3 + w_{14}V_4 + \ldots) + V_2(w_{12}V_1 + w_{23}V_3 + w_{24}V_4 + \ldots) + \upsilon_3(w_{31}V_1 + w_{32}V_2 + w_{34}V_4 + \ldots) + \ldots$$

More explicitly, E may be rewritten as follows:

$$E = -\frac{1}{2}\sum_{i,j} w_{i,j}\upsilon_i\upsilon_j$$

In terms of energy function , the dynamics of the *i* th neuron can be described by (Mehmet and Kamoun,1993)

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i}$$

Where τ denotes a circuit 's time constant. The monotone and increasing function relays the output V_i of neuron i to the entry U_i.

This relation constitutes the foundation of the proposed routing approach.[5]

5. ROUTING IN DISTRIBUTED COMPUTER NETWORKS BY HOPFIELD NEURAL NETWORKS

Routing in a packet switching network consists of determining the best path or route between each node pair (source/destination) through the network in order to minimize the network delay .

A Hopfield network consists of n(n-1) completely connected neurons .Firstly ,the learning of the Hopfield model is static since there is no true dynamism in the connections .Secondly, the relaxation of the network is dynamic since the network is capable of making a certain number of iterations before reverting to a stable state. Another aspect of Hopfield networks is their tendency to minimize an energy functions between the different neurons.

Many new distributed multimedia applications involve dynamic multiple participants, have stringent ends to ends delay requirement and consume large amount of network resources. [2]The neural-network model proposed to solve the routing problem consists of n(n-1) neurons that is, the matrix nx n where all the neurons on the diagonal are eliminated. The coordinates of the neurons are (x,i),where x denotes the rows, and i the columns. The neurons at (x,i) is characterized by its output V_{xi} and defined as follows:

$$V_{xi} = \begin{cases} 1 & \text{If the arc(x,i)is part of the route,} \\ 0 & \text{If not} \end{cases}$$

And the variable ρ_{xi} is defined as follow

$$\rho_{xi} = \begin{cases} 0 & \text{If the arc(x,i)is part of the route,} \\ 1 & \text{If not} \end{cases}$$

The cost of the (x,i) is denoted by C_{xi} which is a real positive variable. A null cost is assigned to each nonexistent arc. For the purpose of numeric manipulation associated with

calculation of the derivatives and without loss of generality ,the energy function proposed by Mehmet and Kamoun(1993) has been adopted

$$E=\frac{\mu_1}{2}\sum_{x=1}^{n}\sum_{\substack{i=1\\ i\neq x\\ (x,i)\neq(d,s)}}^{n} C_{xi}V_{xi}+\frac{\mu_2}{2}\sum_{x=1}^{n}\sum_{\substack{i=j\\ i\neq x\\ (x,i)\neq(d,s)}}^{n}\rho_{xi}V_{xi}$$
$$+\frac{\mu_3}{2}\sum_{x=1}^{n}\Big\{\sum_{\substack{i=1\\ i\neq x}}^{n}V_{xi}-\sum_{\substack{i=1\\ i\neq x}}^{n}V_{ix}V_{ix}\Big\}^2$$
$$+\frac{\mu_4}{2}\sum_{i=1}^{n}\sum_{\substack{x=1\\ x\neq i}}V_{xi}(1-V_{xi})+\frac{\mu_5}{2}(1-V_{ds})$$

Time delay is a function of the link flows and capacities. For high-speed networks, the delay may be obtained by adding the propagation delay (Kleinrock, 1992):

$$T=\frac{1}{\gamma}\sum_{i=1}^{m}\frac{f_i}{C_i-f_i}+\tau$$

Where τ is equal to L/c, L denotes the length of the links, and c the speed of light. If L is expressed in kilometers, then

$$\tau=\frac{10^{-3}}{3}L.$$

6. IMPLEMENTATION AND COMPARISION WITH OPTIMAL ROUTING ALGORITHM

The phenomenal increase of the size of networks and our increasing reliance on them for services requires effective means of network and system management.

In this section, the effect of two parameters (the traffic and the size of the network) on the behavior of the proposed routing method is considered. Consider the network configuration of 10 nodes shown in Fig.3. The node coordinates are given in Table 1. The paths connecting the node pairs 2 –10 are selected for this comparative study. Fig.4.shows the variations of the mean delay as a function of the traffic level. The paths connecting the node pairs 2 –10 are selected for this comparative study. For a uniform traffic of packets/s, an average delay of 59.9ms is obtained for the "shortest distance routing". By this technique, a packet traveling from node 2 to node 10 uses the route 2-1-5-6-10; the related delay is 193.3ms. The path followed from node to node 10 , based on the " minimum number of hops", is always the same(2-4-8-9-10),path which has a length of 4 hops. Fig.4.exposes the mean delays obtained by the neural network method are more stable than those resulting from the other two methods.

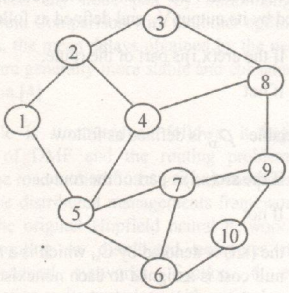


Fig.3. Network of 10 nodes

Table 1 Node coordinates for the network in Fig.3.

Nodes	1	2	3	4	5	6	7	8	9	10
Abscissa	20	20	40	40	30	55	60	70	85	85
Ordinate	60	85	100	70	35	25	60	85	60	30

In order to evaluate the sensitivity of the neural network method relative to optimal routing algorithms. The network configuration of 9 nodes shown in fig.5. has been considered. The comparison results with optimal routing algorithms are reported in Table 2.In spite of the gap that separates the neural network method from those provided by optimal algorithms such as Flow Deviation (FD) (Courtiois and Semal,1980; Fratta and Gerla,1974), and Bersekas-Gallager (BG) (Bertsekas and Gallager,1987;Kershenbaum,1993), this difference is compensated by smaller execution times consumed by the proposed neural-network approach. Obviously, the large number of iterations required by these optimal algorithms before obtaining a solution, as reported in Kershenbaum (1993),causes this contrast in execution times. Compared with optimal algorithms, the proposed

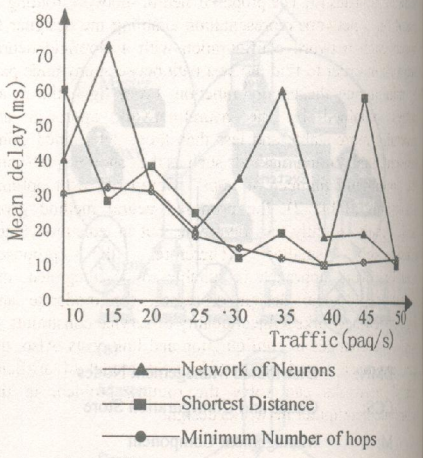


Fig.4. Variations in the mean delay as a Function of the traffic level

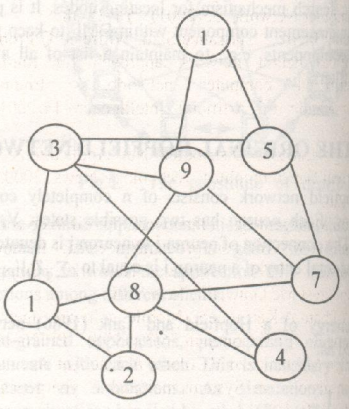


Fig.5. Network of 9 nodes

Table 2 Comparison results of the network in Fig.5.

Routing method	T_{mean}(s)	Execution time(s)
Shortest distance	0.795	0.135
Minimum no.of "hops"	0.450	0.130
Neural network	0.33	0.142
Flow deviation	0.176	5.54
Bertsekas-Gallager	0.180	13.94

Neural-network method gives delay results slightly less favorable, but in execution time(s) considerably smaller.

7. CONCLUSION

In this paper, the distributed management framework (DMF) presented provides an environment that allows a broad range of management tasks to move and run anywhere within the managed system. In our approach, management tasks are lightweight applications that can be configured and downed dynamically as required, reducing the load on managed resources and simplifying the problem of management software updates.[6] The proposed neural -network routing is based on a network representation enabling the designer to match each network configuration with a Hopfield neural network in order to find the best path between any node pair by minimizing the energy function. From fig.4. the mean delays obtained by the neural-network approach are generally more stable and less than those determined using conventional routing methods such as the "shortest path" and the "minimum number of hops". Compared with optimal algorithms (Table 2), the proposed neural method gives delay results slightly less favorable, but in execution times considerably smaller. Therefore, the proposed neural-network approach is suitable to be integrated into overall topological design processes, for moderate and high-speed networks subject quality of service constraints as well as to changes in configuration and link costs. Also, the new approach based on the Hopfield model of artificial neural networks can solve the routing problem in the distributed computer networks design.

REFERENCES

[1]. M.Feridum,J.Krause. Computer Networks, A framework for distributed management with mobile components. Computer Networks 35(2001) PP.25-38.

[2]. Samuel Pierre, Hassance Said, Wilfreied G.Probst. An artifical neural network approach for routing in distributed computer networks. Engineering Application of Artificial Intelligence 14(2001) PP: 51-60.

[3]. Geoge Lawton. Distributed Net Applications Create Virtual Supercomputer. Technology News, 2000.7, PP: 16~20.

[4]. Tivoli Management Framework, Tivoli Web Site: http://www.Tivoli.com/products/index/mgt-framework

[5].Drosen,J., 1994. Neural Network Computing. Windcrest/Mc GrawHill, New York.

[6]. V.A.Pham.A.Karmouch, Mobile software agents: An overview. IEEE Commun.36(3)(1998) PP:26-36

[7]. Dorigo M,Maniezzo V,Colorni A.T he Ant Agents.IEEE Transactions on Systems,man,and C ybernetics—Part B,1996,26(1):1-13

[8]. Schoodnerwoerd R, Holland O,Bruten J et al.Ant-based Load Balancing in Telecommunications Networks.Adaptive Behavior,1997,5(20):169-207

[9]. DeSanctis G, Gallupe R B. A Foundation for the Study of Group Decision Support Systems.Management Science 1987,33(5): 589

[10] . Widom J.Research Problems in Data Warehousing. Proc.4th Intl.CIKM Conf., 1995

* Nature Science Foundation of Hunan Province. (No.OOJJY2082)

A Policy-based Hierarchical Network Management System

Wang Ping, Zhao Hong
Software Center of Northeastern University, Shenyang, 110006, P.R.China
Email: {wangp, zhaoh}@neu.edu.cn

Li Li
Shenyang Institute of Computing Technology,
Chinese Academy of Sciences, P. R. China, 110004
Email: roselily@sict.ac.cn

ABSTRACT

This paper presents a policy-based hierarchical and scalable scheme for distributed network management. The system achieves scalability by modifying the number of components of the hierarchical architecture according to the size of network. The hierarchical approach can efficiently distribute the loads and improve the reliability of the system. In order to increase flexibility, the system employs a policy-based dynamic subscription approach that enables users to manipulate management tasks at run-time. The event-triggered management mechanism enables the managers to discover and solve the problem of network in time. Moreover, the system can trigger or disable management rules automatically according to some network events. This significantly improves autonomy of the system.

Keyword: Hierarchical Architecture, Policy, Domain, Dynamic Subscription, Event-Triggered

1. INTRODUCTION

With the increasing complexity and heterogeneity of modern networks, centralized and weakly distributed network management paradigms show a number of limitations in practice, such as lacking robustness and flexibility. To address this, distributed management paradigm is proposed [1,2]. Goldszmidt with his Management first demonstrated the full potential of large-scale distribution over all managers and agents by Delegation (MbD) framework [3], which set a milestone in this research field. Now, a number of works has been done in the area of distributed network management [4-9]. All of them have overcome the shortcomings of centralized or weakly distributed paradigms in some aspects. However, they are insufficient to support a scalable and flexible task managing mechanism for network management, and do not provide dynamic management capabilities.

The objective of our present study is to improve the scalability and flexibility of the distributed management paradigm. Therefore, we design a hierarchical architecture for distributed network management. It can achieve scalability by modifying the number of components according to the size of network. The management system uses fine-grained decomposition and allocation mechanisms to ensure that the tasks are efficiently distributed among the agents/DMs and prevent event propagation in the network. The administrator can manipulate the management tasks on the fly based on the policy-based dynamic subscription approach. The event triggered management mechanism enables the administrator to solve the problem in the network in time.

The remainder of this paper is organized as follows: Section 2 presents framework of the system; Section 3 describes the management tasks allocation approach; Section 4 explains the management mechanism of the system; Section 5 presents the implementation of the system; Finally, we conclude with perspectives for future work.

2. MANAGEMENT FRAMEWORK

The system adopts a distributed, hierarchical approach for the network management. As illustrated in the Figure1, the overall architecture is organized in at least three levels of hierarchy that may be generalized and expanded horizontally or vertically. The generic architecture includes three core component types:

The management agent operates at the leaf level, lowest in the hierarchy. It is responsible for collecting first-hand, low level information from the managed information entity. The agent gets the status information of managed entities by polling or receiving trap from the SNMP agent. When the status value of managed entity reaches or exceeds the limited, the agent generates event and forwards it to the appropriate DM according to subscription. The agent can be integrated into the managed entity. Because the agent is a lightweight agent and operates with limited system resources, It do not significantly affecting the performance of the managed entity. The Domain Manager operates at between agent and MS and manages a specific domain, acting as a delegate of higher level manager modules. The DM is responsible of detecting network events generated by the agents. A user-interface may or may not be supported by the DM. If no interface exists, then the human operator cannot request directly information and the DM acts solely as a local delegate of higher level manager components. Different DM may execute different management tasks.

The Management System operates at the top level and

This research was partly supported by 863 HTRDP under the contact 863-317-01-08-99

supports the full set of conventional management functions (monitoring, history, alarms, topology, remote configuration, etc.) including a graphical user-interface (GUI).

More levels of DM may be inserted between the MS and agents, thus expanding the hierarchy vertically and generalizing the concept of the domain. The number of levels in the monitoring hierarchy is dictated by the requirements of system functions. For instance, if the function is delay-sensitive, the number of levels should be reduced to minimize the communication latency. On the other hand, if the function needs a high-volume of data and the DMs not residing in powerful machines, the number of levels should be increase in order to help in distributing the monitoring load and alleviate performance bottleneck. In most case, two to three hierarchical levels are sufficient.

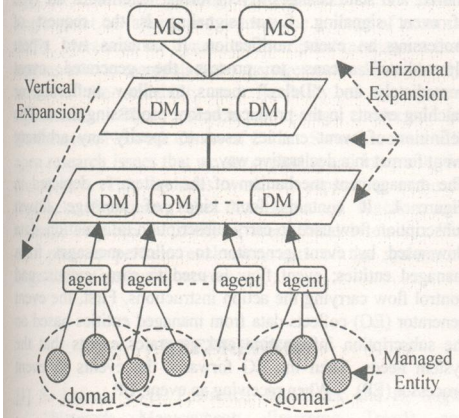


Figure 1: Hierarchical Architecture

Further, multiple MSs may be introduced at the top level (horizontal expansion) for satisfying redundancy requirements or for separating functionality at the top level, e.g. an MS is responsible for planning and global access statistical analysis, another is responsible for fault handling and trouble ticket generation, etc.

3. POLICY-BAESD SUBSCRIPTION MECHANISM

In the system, all management tasks are defined using policies[10-12]. The approach of policy-based subscription mechanism permits the behavior and strategy of the system to be modified by enabling or disabling policies, and replacing existing policies with new ones. Interpreter in manager component interprets policies so that changes can be achieved without shutting down the system. The policy is independent of the manager components and can be reused in different environments. The dynamic subscription improves flexibility of the system.

3.1 Definition

Define 1: Policy is a persistent specification of an objective to be achieved or a set of actions to be performed in the future or as an on-going regular activity. It includes the attributes as follow:

Policy:= <policy-id><subject><target><rule-list>
{<exception>}

The <exception> is optional. The exception mechanism is to catch any problem encountered when initiating a policy. An exception calls a procedure within the manager component to process the problem.

Define 2: Rule describes the management behaviors and can be used by different policies.

Rule := <rule-id> {<trigger>} <action>
{<constraint>}

The <trigger> and <constraint> are optional. If a rule have the <trigger> attribute, the rule will be disabled until the trigger is fired. The system has two type trigger: network event and Timer. The <constraint> attribute limits the applicability of a policy, e.g. to a particular time period. The <action> attribute specifies what must be performed.

Policy subject is the entities to whom the policy is directed and the target is the object at which the policy is directed. The subjects and targets are typically specified as domain [13], not individuals. The member of them can be modified dynamically without affecting the policy.

Define 3: Domain is a collection of member objects which have been explicitly grouped together to apply a common management policy.

Domain: = Object {, Object |, Domain}

The domain is a parent of its members. Since domains are themselves objects, they may be members of other domains, and are then called subdomains of their parent domain. The member of subdomain is called indirect members of the parent domain. Several operators (see Table 1) are defined so that a policy is able to select the set of subject or target objects within a domain to which it applies.

Table 1. Domain Operators

Operators	.
ANY	All member of the domain
+	Add a object to domain
-	Delete a object from domain
∪	Domain union
∩	Domain intersection
*n	A set that contains all direct and indirect members of domain as far down as the n-th level
{a}	Returns the domain that contains the object

3.2 Subscription Mechanism

Subscription is the process that the administrator selects management policy, and decomposes and distributes the rules of the policy to policy subject objects. Policy server, rule server and domain server are set in the system for policy subscription. Figure 2 shows the subscription of a policy.

(1) Administrator selects the policy according to the management tasks and sends the policy-id to policy server. (step①);

(2) Policy server interacts with the domain server to get the address of root subject object and sends the policy to the root subject object. (step②-④);

(3) The root subject gets the target address from domain server and rules from rule server. And then the rule interpreter of the root subject decomposes and distributes the rules to the adjoining low-level management components in the subject domain. This is continuous until the management

tasks are distributed to the management agents, the lowest level in the system. (step⑤-⑧)。

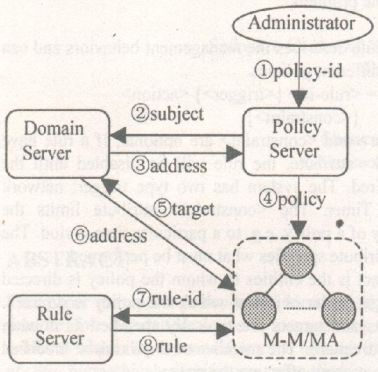


Figure 2: Policy Distributed

The rules have three status: "Dormant", "Enabled" and "Disabled". Figure 3 shows the changes of status during its life. At its conception it acquires its "Dormant" status. Rule keeps this status until the administrator finishes editing it.

Once it is distributed to the management components, the rule will change into "Enabled" status. The status of "Disabled" is set to provided the added flexibility of being able to relatively quickly enable and disable rules at the rule interpreters of the

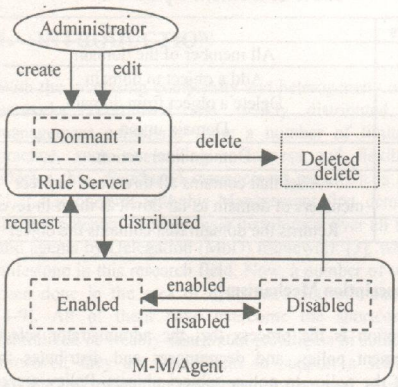


Figure 3: State Transition in the Life of Rule

management components without needing to redistribute them, which could be both time-consuming and costly. When a policy is disabled or replaced with new one, the rules of the policy are not deleted but disabled. If the new policy uses the same rule, it will enable the rule again. When the rule interpreter can not store new rules because lacking of space, the rule that is not used for the longest time will be delete first. Once disabled, a rule can be deleted from an interpreter. The rule still exists in the rule server until it is delete from the server. A restriction on the transitions of rule status is imposed to prevent inconsistencies. The system keeps track of the information for each rule. Only the rule that is not used by any management component can be deleted from the rule server. Then the rule does not actually exist any more in the system.

4. EVENT-TRIGGERED MANAGEMENT MECHANISM

Event is the kernel element and decision-making bases of the system. In the system, the event is classified into two types: simple event is based on a single message and generated by the agent; composite event consists of more than one simple event and is generated by the DM. The definition of an event contains information that captures event characteristics such as event type, event values, event generation time, event source, and state changes. Event format determines the type of event signaling. Event signaling is the request of processing an event notification. It contains two types: "Immediate" means to process the generated event immediately and "Delay" means to allow buffering or batching events in the producer before processing them. The definition of event enables users to specify any arbitrary event format in a declarative way.

The management mechanism of the system is depicted in Figure 4. It contains four kinds of message flows: subscription flow used to carry subscription information, data flow used by event generator to collect messages from managed entities; event flow is used to carry events; and control flow carrying the action instructions. First, the event generator (EG) collects data from managed entities based on the subscription information and generates events that the system needs. Then the EG forwards the events to event processor (EP). When receiving an event, EP

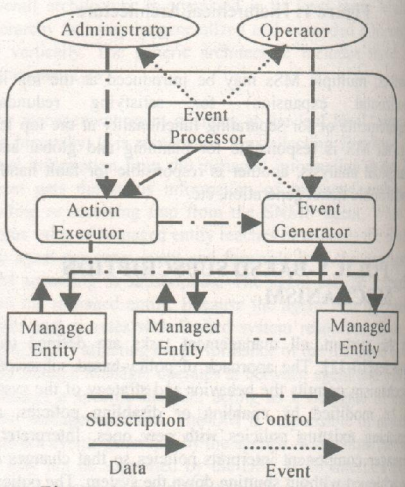


Figure 4: Management Mechanism

checks it according to the rules that stored in database. If EP can process the event, it gives instructions to the action executor (AE) without the administrator. If it can not, EP forwards the event to its top-level manager until to the administrator. If correlated events exist, EP will combine them to composite event and then forward it.

Action in the system is not only simply executing a program or setting a variable. In order to improve the dynamism and

the expressive power, the system provides more complex actions: enabling or disabling rules or subscribing new rules. This enables the system to perform more efficiently. Enabling new rules may also trigger other management operations. This event-action cycle enables system to activate a series of management operations automatically without administrator intervention. The feature significantly improves the dynamism, scalability and performance of the system.

5. CONCLUSION AND FUTURE WORK

This paper presents a hierarchical, scalable and dynamic architecture for network management. Comparing with centralized and weakly distributed paradigms, it improves the performance significantly since the management tasks can be distributed among DMs or/and MSs, avoids single–point of failure, and reduces the amount of network traffic. Moreover, the dynamic subscription mechanism that enables user to manipulate the management tasks on the fly improves the scalability and flexibility of the system significantly. This research is being conducted in Northeastern Network Center of Cernet. Some issues are still open research issues that we are exploring and integrating in the system. This includes work on supporting end-user feedback, fine-tuning the distribution of function and message between the different hierarchical levels etc. We also plan to perform evaluation tests of the system in large-scale distributed network.

REFERENCES

[1] S. Erfani, V. B. Lawrence, M. Malek and B. Sugla, "Network Management: Emerging Trends and Challenges", *Bell Labs Technical Journal*, Oct.- Dec., 1999

[2] J.P. Martin-Flatin, S. Znaty, and J.P. Hubaux. "A Survey of Distributed Enterprise Network and Systems Management". *Journal of Network and Systems Management*, 7(1):9–26, 1999.

[3] Goldszimdt G., Yemini Y., "Distributed Management by Delegation", *15th International Conference on Distributed Computing Systems*, June 1995.

[4] F. Barillaud, L. Deri, and M. Feridun. "Network Management using Internet Technologies". In A. Lazar, R. Saracco, and R. Stadler (Eds.), "Integrated Network Management V". *Proc. 5th IFIP/IEEE International Symposium on Integrated Network Management (IM'97), San Diego, CA, USA, May 1997*, pp. 61–70. Chapman & Hall, London, UK, 1997.

[5] Siegl M.R., Trausmuth G., "Hierarchical Network Management", *In Proc. JENC6*, 1995.

[6] Konopka R., Trommer M., "A Multilayer-Architecture for SNMP-Based, Distributed and Hierarchical Management of Local Area Networks", *4th International Conference on Computer Communications and Networks, ICCCN'95*, 1995.

[7] M. Kahani and H.W.P. Beadle. "Decentralized Approaches for Network Management". *ACM Computer Communication Review*, 27(3):36–47, 1997.

[8] K. C. Budka, R. J. Dipasquale, B. W. A. Rijsman and A. B. Sripad, "Engineering Large SONET/SDH Management Networks", *Bell Labs Technical Journal, Oct. – Dec., 1999*

[9] S. Alexander, S. Kliger, E. Mozes, Y.Yemini and D. Ohsie. "High Speed and Robust Event Correlation". *IEEE Communication Magazine*, pages 433-450, May 1996

[10] R. Wies, "Policies in Network and Systems Management – Formal Definition and Architecture", *Journal of Network and System Management*, Vol. 2, No. 1, Pages 63-83, March 1994

[11] R. Wies, "Using a Classification of Management Policies for Policy Specification and Policy Transformation", *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, California, USA, May 1995

[12] M. Sloman, "Policy Driven Management for Distributed Systems", *Journal of Network and System Management*, Vol. 2, No. 4, 1994

[13] M. S. Sloman, "Domain Management and Accounting in an International Cellular Network", *Third IFIP International Symposium on Integrated Network anagement*, San Francisco, April 1993

Design and Implementation of Communication Software for Computer Cluster*

Gu Zhimin Yao Jiandong
Department of Computer Science and Engineering, Beijing Institute of Technology
Beijing, 100081,China

Zhang Qingfang*
Department of Information Science, Shanxi Finance and Economics University
Taiyuan, Shanxi, 030000,China
E-mail: zmgu@263.net

ABSTRACT

According to ZT-net (the cut-through Network) based on a special cross-switch with wormhole technology, to study a communication software for high performance parallel computing and real-time simulation. The following mechanisms were used in the special message passing library: a special message passing protocol which is not based on TCP/IP, an active message mechanism and a programming polling mechanism. The HPMPS (High Performance Message Passing System) is implemented. The HPMPS system can be used in high performance parallel computing and real-time simulation.

Keywords Computer Cluster; Communication Software; Message Passing Library.

1. INTRODUCTION

The existing LANs are mostly used Ethernet, but the collision detection protocol is low efficiency. With the number of computers on one bus increasing, the probability of collision will increase significantly. In order to expand the network scale, cross-switch can be adopted. But it is expensive with a complex structure. In order to accelerate the communication speed, "wormhole technique" can be used. Because of its high cost, its application has been narrowed to tight coupling parallel computing. It is only used in Dawning in China and Inmos 's ST C104[1] in Britain and Myrinet[2] in USA at abroad. C104 is a wormhole exchanger based on switch-bar with 8B buffer of each channel. Myrinet used the same technique as C104, but both of them are with high cost and complex structure. For example, the message header of C104 used, instead of the output port number, the massive comparative circuits of exchanger to select the output port. So the channel cannot be put through easily, once error occurred, the communication will stopped at once and it has not the function of resending. In order to avoid these technological shortcomings, Professor Kang Jichang put forward the cut-through Network based on the improved wormhole technology. This technology belongs to computer network research field. It can link a cluster of computers into a high-speed LAN and the "end-to-end" communication can be done between computers via the exchanger, which is based on switch-bar. In order to make best use of the hardware facilities and its functions, overcome the overhead of TCP/IP, and bring off the high-speed parallel computing, constructing message passing library which is fit for the cut-through network's characteristic is needed. Thus, user can perform the high performance parallel computing or real-time simulation.

2. NETWORK ARCHITECTURE

An 8-port cut-through exchanger links 8 computers to a cluster computer network-by-network adapter. In the general wormhole technology, the message header and the body were sent together, just like an earthworm, so called it wormhole technology. In this design, the body-head separating technique was adopted, i.e., making the message head separated from the message body. At first, sending the message header to the exchanger independently for requesting a channel. Before the success of request, it occupies no channels so that it will not block the channels during the requesting procedure. Once succeeded, after turning on the corresponding witch-bars, the channel was constructed. When many an exchanger was liked together, the channel can be turned on in turn by sending quite a few message headers. After the channel was constructed, the packages can be delivered to the destination computers directly with only a gate-level delay through one exchanger, the header will turn on the channel in turn. So called what the above mentioned the cut-through network. Controlling the exchanger to request constructing channel, the message header cell (HDR) was adopted; while to cancel the channel, the message end cell (END) was adopted. The Britain Inmos's D-S link was used in the physical channel, i.e. it has two circuit: one is for data transmitting, calling it D (Data); the other is for the Strobe to compensating the clock signal, calling it S(Strobe). D means 1 when it is in high voltage level, while D means 0 when it is in low voltage level. When hopoffing was occurred, the clock signal was created. When the two consecutive codes are 00 or 11, the clock signal was compensated by the hopoffing of the S voltage. The cell can be recognized by D and S's hopoffing simultaneously without

* This paper was supported by the Science & Research Initial Fund of Beijing Institute of Technology (DD9619-1)

the special codes representation, resulting in the cell masking. The exchanger only needs to detect the HDR and END, permitting the packages passing through the turned on channel freely instead of storing in the exchanger temporally. So the architecture of the exchanger can be greatly simplified. The scale of the exchanger can be expanded to 8,16,32- port without difficulties. Although the performance of this exchanger is superior to the general switches, we did not call it "switch" for its simpler architecture.

3. DESIGN AND IMPLEMENTATION OF HPMPS

HPMPS (high performance message passing system) is a distributed computing platform developed by author, which is based on the cut-through network and Windows environment. It includes message passing system call and run-time supporting environment. Making best use of the hardware function of the cut-through network, HPMPS can do the massage construction, point-to-point communication, global operation and active message etc. through message passing system library.

3.1 System Configuration

The mp_lib is message passing system call library in the configuration of HPMPS. In order to improve the efficiency, the following designing rules were adopted:

(1) Creating the message communication primitives on the cut-through network protocol instead of TCP/IP;
(2) Introducing the ACTIVE MESSAGE, which supports high-layer communication mechanisms. It can reduce the communication throughput and cost;
(3) Offering the broadcasting mechanism based on hardware;
(4) Introducing the programmable POLLING mechanism in order to reduce the interrupt cost and communication balking.

3.2 Message Passing Call Library

The message passing library mainly includes the functions of message construction , point-to-point communication and active message etc.

(1). Message construction
Message construction is packaging the local machine's data with types preparing to send to the channel, and depackaging the data from the receiving buffer into the typed data. Packaging function packages the data to the sending buffer and returns the pointer which points to the buffer. Depackaging function depackages the data extracted from the receiving buffer into the typed data and returns the pointer which points to the next data position in the buffer.

(2) Point-to-point communication
Point-to-point means the procedure of message passing between two machines. It can be classified into 4 types: write-no-balking and read-balking; both read and write-no-balking; write-balking and read-no-balking and both read and write-balking. Considering the single-process factor, only two of them were designed, i.e both write and read-balking and write-balking and read-no-balking. In addition, the new POLLING type function mp_probe() was introduced.

(3) Global operation
① Broadcasting (supported by hardware): broadcasting the message onto the cut-through network.
② Synchronism: the barrier synchronous function can be created by receiving function、broadcasting function and sending function.

(4) Active Message
After receiving message, this function goes into handler processing and returns after finished the task. RPC(Remote Procedure Call) mechanism can be constructed using this function.

3.3 Implementation Technique

(1) Packaging/Depackaging
As far as packaging is concerned, copying all kinds of types of data according its length to the sending buffer using MEMMOVE function. While the depackaging is restoring the data of MAILBOX buffer into its original type in turn.

(2) Point-to-point communication

[Algorithm 3-1] The basic algorithm for sending data. Aim: sending what the mp_str points in the buffer from src to destination obj.

STEP1: change obj into hdr address, mp_str1=mp_str; ZT network addressing, drawing the message length ml(bytes) in the buffer and modifying ml into multiple of 4.

STEP2: waiting for the ACK which is returned by addressing, If succeeded, go to STEP3.

STEP3: if ml≤512, sending the data pointed by mp_str1, sending EOM. Go to STEP9.

STEP4: else if ml>512, then send 512 bytes data from the buffer pointed by mp_str1,Send EOP, ml=ml-512, modifying mp_str1 to point to the data which will be sent the next time.

STEP5: waiting for ACK, After received it, go to STEP6.

STEP6:if ml≥512, then send 512 bytes data from the buffer pointed by mp_str1,send EOP, ml=ml-512,modifying mp_str1 to point to the data which will be sent the next time. If ml<512, go to STEP8.

STEP7: if ml≠0, then waiting for ACK, if arrived, then go to STEP6.

STEP8: if ml≠0, then send the rest data, send EOM.

STEP9: waiting for ACK, if arrived, then returns 0,exit.

[Algorithm 3-2] basic receiving function r-recv1(). Aim: extract the current message from the channel and put it into the MAILBOX.

STEP1: waiting for EOP/EOM, if arrived, go to STEP2.

STEP2: extract the first 4 bytes and then change it into typed data ml(message length).

STEP3: determine the memory spaces based on the requesting receiving buffer of ml and let mp_str3 point to its beginning address, mp_str2=mm_str3.

STEP4:if ml≤512, then extract data from the channel and then put it into the buffer where mp_str2 points to. Modify mp_str2 to point to the end of the data. Send ACK. Go to STEP8.

STEP5: when ml≥512, extract data from the channel and put it into the buffer pointed by mp_str2. Modify mp_str2 to point to the end of the data. Send ACK.
ml=ml-512. if ml<512 then go to STEP7.

STEP6: waiting for EOP/EOM, if arrived, go to STEP5.

STEP7: receiving the rest of data from the buffer and put them into the buffer pointed by mp_str2. Modify mp_str2 to point to the end of the data. Send ACK.

STEP8: save mp_str3 into MAILBOX, reset the status and

exit.

As for the function mp_recv(mtype), it extracts the message which matches with mtype(message type). Read-balking function mp-brecv(mtype) is constructed by mp-recv() and r-recv1(). While function mp-strobe used for probing whether there are data arrives or not, if there, then call r-recv1(), else return –1.

(3) Global operation

As far as broadcasting mechanism is concerned, it is supported by hardware. When the addressing is to itself, the sending of local machine is broadcasting with the function implemented by mp_bcast(mtype). As for barrier mechanism, the aim is to satisfy the synchronous need among group of tasks. If it is a console, receiving the messages from the other nodes first and count. After all messages were received, it broadcasts another message and exit. If it is a node, send message to console first, if it received message which means the end of synchronism and exit.

(4) Active message

Aim: after receiving message, go to handle processing.

The main points of algorithm: after receiving message using mp_brecv(), execute the function handle which should be defined by user. The function RPC can be implemented by this mechanism.

4. ANALYSIS

First of all, analyze this design from the point of view of theory. Robin Milner [5] thought that: as for concurrent computing, there exist not only one conceptual model or only one popular architecture. There are different languages、calculus and theories according to different specified fields. Try to find a public semantic frame to link and organize many a interpreting layers. It is well known that concurrent computing is all our research subjects while sequencing computing is only a special field with good performance among it. According to the sequencing computing , the λ -calculus of Church is a famous prototype. The domain theory of Dana Scott acts as the semantics for λ -calculus. Plotkin used power domain theory to deal with the non-definitive questions resulted from the concurrency. Concurrent computing must include concurrent mechanism 、interacting mechanism and variables localizing mechanism. In HPMPS, this basic requirements were satisfied by point-to-point communication、global operation and C++ supporting through MPMD model programming.

From the point of view of efficiency, in general, for the traditional send/receive operation, buffer management is needed, in addition to the protocol processing, the cost is large. However, the key technique of Active message mechanism is substituting the buffer management for chip-level registers, so the cost is small. In order to reduce the time spending in waiting for signal during the communication protocol procedure, two kinds of methods were utilized. The first is interrupt; the other is POLLING. If the message length is short with short interval, the network congestion may be resulted from using interrupt technique. The reason is the much time consumed by the following: the system status saving and restoring which is incurred from interrupt; interrupt opening and closing; and interrupt processing. On the contrary, if the message length is long with long interval, using interrupt technique will not resulting in network congestion. However, when dealing with short message using POLLING technique, because CPU does no exception processing, the network congestion can be avoided by receiving the message in time. As for the messages with long interval, the cost may be increased with the number of detecting rising.

Trade-off: during the parallel/distributed programming, POLLING technique should be adopted for the short message with short interval. So we introduced the function mp_probe to implement this function. As for the active message, the following amelioration was done: considering that resending is needed after data error and constructing high-level communication primitives freely, we can reserve small BUFFER defined by user.

5. CONCLUSIONS

Concurrent computing must include concurrent mechanism、interacting mechanism and variables localizing mechanism. In HPMPS, these basic requirements are satisfied by point-to-point communication、global operation and C++ supporting through MPMD model programming. From the point of view of efficiency, when designing system call function library, POLLING technique should be adopted for short message with short interval. We introduced function mp_probe to implement this function. As for the active message, considering that resending is needed after data error and constructing high-level communication primitives freely, we can reserve small BUFFER defined by user.

The implement technology studied in this paper can be used in high-speed parallel computing and real-time simulation. This system can be improved further. Until now, I/O type is used for the data transmitting between main board and network adapter. It fits for transmitting small message because it takes up CPU time. While for the large message, DMA should be adopted.

REFERENCES

[1] The ST C104 Datasheet. SGS-THOMSON Document No. 42-1470-04.1994

[2] Mario Gerla, Prasasth Palnati, Simon Walton. Multicasting Protocols for High-Speed Wormhole Routing Local Area Networks.

[3] Sunderam V. PVM:A framework for parallel distributed computing. PVM Software Package,1995

[4] Kim W, Lochousky F H. Object-oriented concepts, databases and applications. ACM Press.1989

[5] Miner R. Basic for interacting .Computer Science,1994,21(3):pp1-8

Comparing the Performance of the Cooperative Web Cache System

Bin Xuelian[1] Yang Yuhai[2] and Jin Shiyao[1]
[1]Institute of Computing Technology, National University of Defense Technology,
Changsha 410073, P.R. China
[2]Group of Graduate, AFRA
Wuhan 430010, P.R. China
E-mail: binxuelian@yeah.net

ABSTRACT

Sharing files cached among web proxies is an important method to reduce the traffic over Internet and alleviate network bottlenecks. In this paper, we propose a new cooperative web cache system (HMCS) based on hybrid management after analyzing the existing cooperative web cache system. Different from CRIPS and ICPS systems, HMCS uses redirecting approach that is if the requested file misses in a proxy, then the proxy will redirect it to the primary proxy which caches it and the primary proxy will return it to client directly. By this way, the duplicated file number can be reduced and the efficient storage usage of the proxy system can be increased. Consequently, the hit rate will increase. Theoretic analysis and simulation show that the performance of HMCS is much better than that of CRISP and ICPS system.

Keywords: Central Manager, Distributed Manage Ment, ICPS, CRISP, Cooperative Cache System.

1. INTRODUCTION

Caching popular objects close to the clients has been recognized as the one of the effective solutions to alleviate web server bottlenecks, reduce traffic over the Internet and improve the scalability of such a system. But because users tend to access many sites, each for a short period of time, hit rates of per-user caches are low. Therefore, some organizations have begun to utilize shared proxy caches so that each user can benefit from data fetched by others. Though CRISP and Squid are widely used cooperative cache systems, there are duplicated files in them. Maybe for a period of time, every proxy caches a popular file! Thus limits the collaboration among proxies and leads to the inefficient usage of storage. In this paper, we propose a Cooperative Web caching System based on hybrid management. It can reduce the number of the duplicated files.

The rest of the paper is structured as follows. Section 2 depicts the CRISP system and the Squid system based on ICP protocol and proposes HMCS architecture. Section 3 evaluates system performance and section 4 presents the simulation results. Finally, section 5 summarizes our conclusions.

2. COOPERATIVE CACHE

In this paper, we classify cooperative caching system by the mode of management, split them into central, and distributed management.

Cooperative cache systems based on central management

The representative cooperative cache systems based on central management is CRISP. Figure 1 depicts the structure of a CRISP cache.

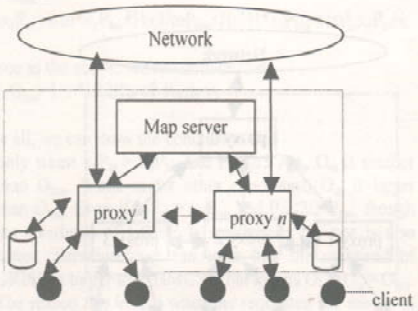


Figure 1 CRISP proxy cache

A CRISP cache consists of a group of cooperating caching servers sharing a central directory of cached objects. Each client is bound to one of several caching servers or proxies, which cache objects on behalf of their clients. Any URL fetch request that misses in the local proxy is send to the mapping server to probe the cooperative cache. The mapping server maintains a complete cache directory; proxies notify the mapping server any time they add or remove an object from the cache. In this way, any proxy can probe the entire cache with a single nicest message exchange. If the map indicates that a requested object is resident in a peer, the requesting proxy retrieves the object directly from that peer, and returns it to the client.

CRISP uses unicast, which alleviates the traffic and reduces the latency. But the map server can be a bottleneck. Moreover, the map server can be a single failure. When map server fail, all the proxies in CRISP can not collaborate.

The cache systems based on distributed management

Squid and Netcache are the widely used cache systems. Their management is distributed on proxies. By this way, there is no single failure.

Squid and Netcache use the ICP protocol to probe peers whether they have the requested files that miss in an allocated proxy. So we call Squid and Netcache are ICPS. When the URL requested missed on a proxy, it broadcasts the message to its peers. If the one peer has the requested file, then the proxy fetches the file from it. If none of the peers have it, the proxy must wait until all peers answer 'HIT-MISS' before send the request to the parent. Figure 2 depicts the ICPS

proxy cache.

ICP reduces the traffic between the proxy and web server, but multicasting increases network traffic and forces all caches to respond to each request at most cases.

Both of the above cooperative cache systems have a common drawback that a lot of duplicate files are cached in the cache systems because the proxies retrieve and cache the file missed in it no matter whether their peers have cached the file. Therefore maybe for a period of time, popular files are cached in many of proxies. This result in inefficient usage of storage and the cooperation among proxies are very limited.
In order to overcome the shortcomings of the two kinds of cache systems and improve the efficiency of storage, we provide a new cooperative cache architecture based on hybrid management.

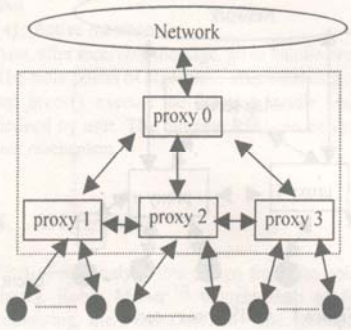


Figure2 ICPS proxy cache

Cooperative cache system based on hybrid management

The HMCS' structure is very similar to CRISP's. Figure 3 depicts the structure of HMCS. HMCS has a manager and its function is same as CRISP mapping server. The main difference between HMCS and CRISP is that proxies in HMCS maintain partial information of files cached in the peer proxies. By this way, the workload on manager and the traffic between proxies and manager reduce and if the manager fails, proxies in HMCS can be limited cooperation, increasing availability. In the paper, the proxies which clients allocate are called master proxy and the proxy that caches the requested file is called primary proxy.

Each proxy has a unique identity and a redirector. Redirector records other proxies' identity and their cached files. If a requested file misses in master proxy, it searches the redirector to probe if there is a primary proxy caching it. If it hits in the master proxy's redirector, then the request is forwarded to the primary proxy. The primary proxy will send the client packets directly without going through the master proxy. If a requested file neither hit in the master proxy nor the redirector, the master proxy communicates with the manager to probe whether it has been cached in the cache system. If it hits in HMCS, the manger returns the primary proxy's id. The master proxy will record the id in redirector and redirect the request to primary proxy. If it misses in HMCS, the manger will return the miss information and the master proxy will retrieve it from web server. Proxies notify the manager any time they add or remove an object from the cache. Using the redirecting approach, the number of duplicated file and the working load on the manager are reduced. Consequently the usage of storage and the hit rate are increased. Moreover, the traffic between proxies and manager reduce and if the manager fails, proxies in HMCS can be limited cooperation increasing the availability.

3. ANLYZING THE PERORMANCE OF THE COOPERATIVE CACHE SYSTEM

This section will analyze and compare CRISP, ICPS and HMCS about the hit rate, the overhead of process and latency. Suppose the number of proxies of the three cache systems is m. the storage and the capability of the proxy process are same. The sequence of the client request is also same.

Hit Rate

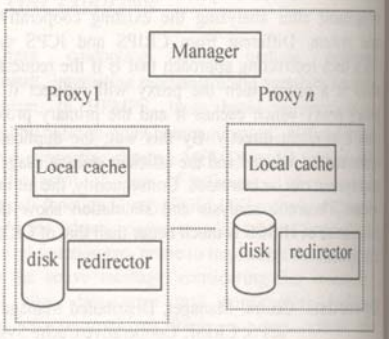


Figure 3 structure of HMCS

Let P_{hm} ,P_{icp} and P_{cr} denote the hit rate of the HMCS, ICPS and CRISP.

Form section 2, we can infer that when a request file misses in the allocated proxy but hits in CRIPS and ICPS, the proxy will retrieve and cache it. This will cause high redundancy. While in the HMCS, the master proxy will forward the request to the primary master not caching the requested file. So the redundancy of HMCS is less than that of CRISP and ICPS and the efficiency of storage of HMCS is larger than that of CRISP and ICPS. Therefore, $P_{hm}>=P_{icp}$, $P_{hm}>=P_{cr}$.

The overhead of CPU processing

Considering a request q, q will be send to one proxy of the system. The probability of the client allocates proxy to P is 1/m. Because P_{hm}, P_{icp} and P_{cr} is hit rate of cache systems, the hit rate of the single proxy can't exceed that of the system. Let x_1P_{hm}, x_2P_{icp} and x_3P_{cr} ($1/m<=x_1,x_2,x_3<=1$) denote hit rate of the proxy P. Assume H is the overhead of receiving or answering a HTTP request. H_{pq} is the overhead of transmitting among proxies. H_{pm} is overhead of transmitting between the proxy and the manager; I is the overhead of ICP information including the overhead of searching, hitting, missing and answering.

Now, we investigate the overhead of a proxy P process.
As for HMCS, if q is firstly send to P, the following is the process. If q hits in P, the overhead of P is 2H including the overhead of receiving and answering a HTTP request. If q misses in P but hits in redirector, q will be redirected to the

primary proxy, the overhead of P is $H+H_{pq}$ or $H+H_{pm}+H_{pq}$ (client firstly request P the file). If q misses in the cache system, after communicating the manager, P will retrieve the file from web server, the overhead of P is $4H+H_{pm}$ including the overhead of retrieving the file from web server, receiving and answering a HTTP request and communicating the manager. If q is firstly send to other proxy but hits in P, then the overhead of P is 2H.

So the average overhead of P processing of HMCS is

$$O_{hm}=1/m[x_1P_{hm}*2H+(1-P_{hm})*(4H+H_{pm}) +(1-x_1)P_{hm}*(H+H_{pq})]+(m-1)/m*2H*x_1P_{hm}$$

Omitted the transmitting overhead H_{pq} because request message is so small, it is simplified as formula (1)

$$O_{hm}=4/m*H+2H*x_1P_{hm}-1/m*3P_{hm}*H-1/m*x_1P_{hm} \quad (1)$$

As for ICPS, if q is firstly sent to P and hits in P, the overhead of P is 2H. If q is firstly sent to P but misses in P, then P will broadcast ICP message and retrieve it from a peer or the web server, the overhead of P is 2I(m-1)+4H. If q is firstly sent to other proxy but misses in it, then P will receive and answer an ICP request. If P misses it, the overhead of P is 2I. If it hits in P, P will receive and answer an extra HTTP request, so the overhead of P is 2I+2H. So the average overhead of P process in ICPS is

$$O_{icp}=1/m\{P_{icp}*2H+(1-x_2)P_{icp}[2I(m-1)+4H]+(1-P_{icp})[2I(m-1)+4H]\}+(m-1)/m[(2I+2H)*(1-x_2)P_{icp}+(1-P_{icp})*2I]$$

Simplifying it as (2)

$$O_{icp}=4/m*H+2H*x_2P_{icp}-4/m*x_2P_{icp}+4I(1-P_{icp})(m-1)/m \quad (2)$$

As for CRISP, if q is firstly sent to P and q hits in P, the overhead of P is 2H. If q is firstly sent to P but misses in P, the overhead of P is $4H+H_{pm}$ including the overhead of communicating with manager and retrieving it from a peer or web server. If q is firstly sent to other proxy but hits in P, the overhead of P is 2H including the overhead of receiving and answering an HTTP request.

So the average overhead of P processing in CRISP is

$$O_{cr}=1/m[2H*x_3P_{cr}+(4H+H_{pm})*(1-P_{cr})+4H+H_{pm}*(1-x_3)P_{cr}] +(m-1)/m*2H*x_3P_{cr}$$

Omits the overhead of H_{pm} and simplifies it as formula (3)

$$O_{cr}=4/m*H+2H*x_3P_{cr}-4/m*x_3P_{cr} \quad (3)$$

Because both of ICP and CRISP are using copying method no matter whether they have cached the request file or not, the redundancy and hit rate is similar by and large.

Comparing formula (2) and (3) we can deduce that formula (2) is $4I(1-P_{icp})(m-1)/m$ larger than formula (3). Therefore, $O_{icp}>O_{cr}$.

Now, we will compare HMCS and CRISP.

If $x_3P_{cr}=x_1P_{hm}$, the result of formula (3) subtracting formula (1) is $Ocr-O_{hm}=3/m(P_{hm}-x_1P_{hm})$.

For $1/m<=x_1<=1$ and $x_3P_{cr}=x_1P_{hm}$, the result of $O_{cr}-O_{hm}$ is larger than 0.

If $x_3P_{cr}>x_1P_{hm}$, the result of formula (3) subtracting formula (1) is

$$O_{cr}-O_{hm}=2Hx_3P_{cr}+1/m*x_1P_{hm}*H+3/mP_{hm}*H-H*x_1P_{hm}-1/m*x_3P_{cr}H>2Hx_3P_{cr}+1/m*x_3P_{cr}*H+3/mP_{cr}H-4/m*x_3P_{cr}H-H*x_1P_{hm}$$

For the redundancy of CRISP is larger than that of HMCS and $x_3, x_1>=1/m$, $O_{cr}-O_{hm}>1/m*H(5P_{cr}-2P_{hm})$;
We can deduce that

if $P_{cr}>2/5\ P_{hm}$, then $O_{cr}>O_{hm}$; if $P_{cr}<2/5\ P_{hm}$, then $O_{cr}<O_{hm}$.

If $x_3P_{cr}<x_1P_{hm}$, the result of formula (3) subtracting formula (1) is

$$O_{cr}-O_{hm}=2Hx_3P_{cr}+1/m*x_1P_{hm}*H+3/mP_{hm}*H-2H*x_1P_{hm}-4/m*x_3P_{cr}H.$$

As same as the above, we can deduce that
$O_{cr}-O_{hm}>1/m*H(2P_{cr}+2P_{hm})>0$.

Above all, we can draw the conclusion that only when $x_3P_{cr}>x_1P_{hm}$ and $P_{cr}<2/5\ P_{hm}$, O_{cr} is smaller than O_{hm}. While under other conditions, O_{cr} is larger than O_{hm}. Even if $x_3P_{cr}>x_1P_{hm}$ and $P_{cr}<2/5\ P_{hm}$, though the overhead of CRISP is smaller, its hit rate is also lower. Therefore, we can infer that the overhead of CRISP is larger than HMCS. That means $O_{icp}>O_{cr}>O_{hm}$. The reason lies in that when the requested file misses in the allocated proxy but hits in the cache system, the approach of the cache system adopted is different. In the ICP and CRISP, the allocated proxy uses copying method, while in HMCS, the allocated proxy uses redirecting method. The decreased the number of copy will improve the performance greatly.

Latency

Considering the latency relationship among client C, cache systems and web server. Latency is the delay between the time that a client requests a file and receives the requested file. Let D_{cp} denote the HTTP delay (requesting/ answering) between the cache system to client C. D_{pp} denotes the HTTP delay (requesting/answering) between proxy and web server. D_{pm} denotes the transmitting delay between proxy and manger. D_{pq} denotes the transmitting delay between proxies.

As for HMCS, there are cases of a requested file hitting or missing in the cache system. If the requested file misses in HMCS, then the master proxy will communicate manager and retrieve it from web server, so the latency is $D_{cp}+D_{ps}+D_{pm}$. If it hits in HMCS and the master proxy, then latency is D_{cp}. If it hits in HMCS but misses in the master proxy, the latency is $D_{cp}+D_{pq}$ or $D_{cp}+D_{qm}+D_{pm}$(client firstly requests the file to the master proxy). Therefore latency is

$$L_{hm}=D_{cp}*x_1P_{hm}+(1-x_1)P_{hm}(D_{pq}+D_{cp}+D_{pm})+(1-P_{hm})(D_{ps}+D_{cp}+D_{pm})$$

or $$L_{hm}=D_{cp}*x_1P_{hm}+(1-x_1)P_{hm}(D_{pq}+D_{cp})+(1-P_{hm})(D_{ps}+D_{cp}+D_{pm})$$

Generally speaking, cache proxy belongs to a same ISP, the transmitting delay between proxies and manager is very little and it can be neglected. We can simplify the above formula as

$$L_{hm}=D_{cp}+(1-P_{hm})D_{ps} \quad (4)$$

As for ICPS, we should also consider the ICP delay (requesting/answering) among proxies. Let D_{icp} denote the ICP delay. If the request file hits in the allocated proxy, the delay is D_{cp}. If the allocated proxy misses the requested file, it will broadcast ICP message to peers to get the information whether the peers cache it and retrieve it from a peer or web server, so the delay is

$$D_{icp}+D_{pp}+D_{cp} \text{ or } D_{icp}+D_{ps}+D_{cp}.$$

Therefore the latency of ICPS is

$$L_{icp}=D_{cp}*x_2P_{icp}+(1-x_2)P_{icp}(D_{icp}+D_{pp}+D_{cp})+(1-P_{icp})(D_{icp}+D_{pp}+D_{cp})$$

Or

$$L_{icp}=D_{cp}*x_2P_{icp}+(1-x_2)P_{icp}(D_{icp}+D_{pp}+D_{cp})+(1-P_{icp})(D_{icp}+D_{ps}+D_{cp})$$

The latency of ICP can be omitted because ICP transmitting is based on UDP and UDP latency is very little. So the above formulas can be simplified as

$$L_{icp} = D_{cp}+(1-x_2P_{icp})D_{pp}+(1-P_{icp})D_{ps} \quad (5)$$

As for CRISP, if a requested file misses in the cache system, the allocated proxy will communicate manager and retrieve it from web server, so the delay is $D_{pp}+D_{cp}+D_{pm}$. If it misses in the allocated proxy but it in CRISP, the allocated proxy will retrieve it from a peer, so the delay is $D_{cp}+D_{pm}+D_{pq}$. If it hits in the allocated proxy, the delay is D_{cp}. Therefore the latency of CRISP is

$$L_{cr}=D_{cp}*x_3P_{cr}+(1-x_3)P_{cr}(D_{pp}+D_{cp}+D_{pm})+(1-P_{cr})(D_{ps}+D_{cp}+D_{pm}).$$

It can be simplified as formula □, omitted the delay between proxies and the manager.

$$L_{cr}=D_{cp}+(1-x_3P_{cr})D_{pp}+(1-P_{cr})D_{ps} \quad (6)$$

Because $P_{hm}>P_{cr}$ and the redundancy of HMCS is smaller than that of CRISP, comparing formula □ and formula (6), we know $L_{hm}<L_{cr}$. Supposing the hit rate and redundancy of CRISP and ICPS is same, we can get $L_{cr}=L_{icp}$. So $L_{hm}<L_{icp}$.

Above all, we can learn that the latency of HMCS is less than that of CRISP and ICPS because the redundancy of HMCS is less than that of CRISP and ICPS, resulting in that HMCS' hit rate is higher than CRISP and ICPS's.

4. THE SIMULATION RESULTS

In this paper we uses Wison Proxy Benchmark1.0 [6] to simulate the HMCS, ICP and CRISP. All the three cache systems adopt LRU replacement. Because there are not many PCs available for us, we rewrite Wison Proxy Benchmarking1.0 so that it can run in one PC. The results we can get are the hit rate and the number of messages. The time of CPU process is not available for us. Though we can get the overhead of CPU process using Wison Proxy Benchmark1.0, it also cannot reflect the real overhead of CPU process. So we are sure that though we use just one PC to simulate, the results could reflect the hit rate and the number of messag that cache systems fork. We use the number of message t represent the overhead of CPU process. Table 1 and table describe the results of the simulation for 20 minutes.

From table 1 we can learn that the hit rate of HMCS i between 6% and 7% higher than that of ICPS and CRISP From table 2 we can learn that under the same reques number conditions, the extra message of HMCS is the leas that of CRISP is the less and that of ICP is the most. Th workload on the manager of the HMCS is 10% smaller tha that of CRISP. Therefore we can conclude that th performance of HMCS is much better than that of CRISP an ICPS.

Table 1 the hit rate of HMCS, ICPS and CRISP

Hit rate	HMCS	ICPS	CRISP
Cache system hit rate	49.91%	42.1%	42.9%
Local proxy hit rate	12.5%	12.5%	12.6%
Peer hit rate	37.5%	29.7%	29.9%
Cache system miss rate	49.98%	57.9%	57.1%

Table 2 the overhead of HMCS, ICPS and CRISP's process (The value of table is the number of message divides the tota number of message.)

	HMCS	ICPS	CRISP
Local proxy	48.42%	38.74%	47.51%
Peer proxy	18.13%	61.27%	9.86%
Manger	33.44%	0	42.58%
The number of Client send request	48.42%	30.53%	47.51%

5. CONCLUSION

Before designing the web cache systems, we analyze the existing cooperative cache systems in detail. We think that the hit rate of both the cache systems based on central and distributed management is very low. The reason lies in that the number of duplicated files in both of the systems is very large resulting in the inefficient usage of storage and limited collaboration. In order to overcome the above shortcomings we propose a new cooperative cache system HMCS based on hybrid management. HMCS reduces the redundancy increases hit rate and improves the performance of cache system, using redirecting. Furthermore, when the manager fails, the proxies are limited cooperation, increasing availability. From analysis and simulation, we draw the conclusion that the performance of HMCS is much better than that of ICPS and CRISP.

REFERENCES

[1]Peijung Ho, Florin Baboescu. Cooperative Proxy Caches-When all is said and done. http://www-cse.ucsd.edu/~baboescu/research/research.html

[2]Wessels D, Claffy K. ICP and the Squid Web Cache. IEEE Journal on Selected Areas in Communication, 1998, 16(3) pp.345~357

[3]CRISP: A Cooperative Internet cache. http://www.cs.duke.edu/ari/cisi/crisp
[4]LIN Yong-Wang, ZHANG Da-Jiang and QIAN Hua-Lin. A Cooperative Web Caching System Based on Concentrated Management. Journal of Computer Research & Development, 2001, 38(1). pp.63~73
[5]ZHENG Xiao-Wei, ZHENG Wei-Min and SHEN Mei-Ming. A Distributed Cooperative Caching Algorithm for Workstation Cluster File System. Journal of Computer Research & development, 1999,36(9), pp.1057~1061
[6] Wison Proxy Benchmark1.0. http://www.cs.wisc.edu/~cao/wpb1.0.html

A Priority Buffering Queue Architecture in High-Speed Switch

Yang Yuhai
Group of Graduate, Airforce Army Radar Academy
Wuhan, Hubei 430010, P.R.China
Bin Xuelian
Group of Doctor, Institute of Computing Technology, NUDT
Changsha, Hunan 410073, P.R.China
Zheng Yuqiang
P&S Electronics CO. Ltd
Wuhan, Hubei 430070, P.R.China
E-mail: yyhandy@yeah.net

ABSTRACT

It is evident that an efficient buffering and queuing scheme play a key role in the performance of the switch. In order to make the schedulers flexible and effective and guarantee quality of service (QoS), We need to design a buffering queue architecture that supports priority scheduling. In this paper, we propose priory FIFO queue architecture PFQ after studying conventional buffering and queuing techniques. It is based on the basic idea of shift register. Each priority FIFO queue is organized as a linked list. It solves the HOL (Head of Line) problem efficiently by setting the high-speed local bus. Simulation results show that PFQ is highly flexible and efficient at a low hardware expense. Furthermore, using PFQ in the switch, we can implement a much more flexible scheduling algorithm. We present a scheduling algorithm PSLP, which is very easy to implement. At last, we draw a conclusion that PFQ can be used in guaranteeing QoS requirements in high-speed switch and can be designed as a single-chip to simplify switch design.

Keywords: FIFO, Priority, Buffering and Queuing, PFQ, Priority Scheduling.

1. INTRODUCTION

A crossbar switch is the core of many routers nowadays. A switch's job is to receive packets at its input ports and forward them to its output ports. When two packets destined for the same output port arrive at different input ports of a switch at approximately the same time, they cannot both be forwarded immediately. Only one packet can be transmitted from an output at a time. Hence one of the two packets must be buffered for later transmission. Otherwise it may be lost. Crossbar switch widely employs the buffering and queuing technique to optimize performance. So the architecture of the buffering queue has important effect on the performance of switch. In this paper, we propose an input buffering queue architecture supporting QoS, after studying the widely used methods for buffering and queuing.

There are many implementations of buffering queue including FIFO and VQ. In the paper, we illustrate buffering and queuing at the input port to analyze and design, and assume that packet will be segmented to several cells when it arrives at the switch.

FIFO is a queue that buffers cells according to their order of arrival, as shown in figure 1.

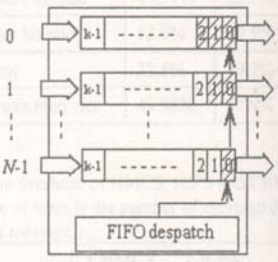


Figure 1 FIFO mechanism

Only the first arrival cell, the head of queue, has the chance to be scheduled. Although the implementation of FIFO is very simple, the drawback is that the head cell that can't be scheduled will block other cells sent to the free port due to the output port contention. This problem is known as HOL(Head of Line). HOL will decrease the bandwidth utilization and limit the throughput of the switch. Reference [1] analyzed this phenomenon and pointed out that the bandwidth utilization of the crossbar is only 58.6% or even lower. Then, some researchers proposed a method to avoid HOL. It uses virtual queue (VQ) scheme to organize the buffering queues.

When VQ is used at the input port , cells are grouped into different queues, according cells' output port. It is called virtual output queue (VOQ). It is shown in Figure 2

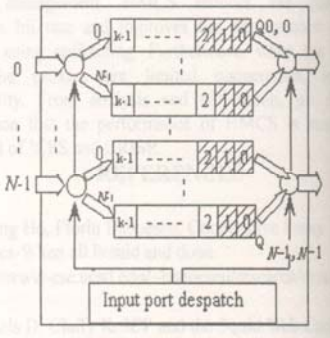


Figure 2 VOQ mechanism

$Q_{i,j}$ means the queue holds cells sent from i input port to j output port ($0 \le i,j \le N-1$). There is no HOL because different VOQ queues have different output port. Using specific scheduling algorithm, it can obtain 100% switch bandwidth utilization. Nowadays, there are many commercial routers apply VOQ mechanism to organize switch's input queue. Although VOQ solves the problem of HOL quite well under the condition of few number port, the utilization of resources is very low. At most cases, a large amount of buffer is free. Furthermore, with increasing ports, the required buffers will increase by O (N^2), which weakens scalability. Therefore, VOQ is only suitable for the switch with few number ports.

2. A PRIORITY BUFFERING QUEUE ARCHITECTURE

IETF presented two Internet QOS model---Diffserv and IntServ. Here we only research how to support Diffeserv. In order to support Diffserv, priority must be used in the switch. Assume the number of priority is p. As for p, it needs at least $\log_2 p$ bits of a cell. Using this mechanism, highest-priority cells will be scheduled at first. Because the length of most packets through the network is short and the transmitting speed is very high, just using software solutions does not meet the speed requirement for software solutions are not fast enough to keep up with the packet transmission rate. For example, in a 2.5Gbps ATM network, an ATM cell can be transmitted in 0.1696us. Within the period of 0.1696us, switch must do two things. One is enqueuing a new incoming cell, the other is dequeuing the highest-priority cell. There are some extra overhead using software solutions, for instance, retrieving instruction, decoding instruction, processor responding and sending data. This will lead to low transfer speed. While a hardware solution can operate at the speed of closing to the operating speeds of the link. Moreover, a hardware solution can overlap enqueue and dequeue operations to avoid wasting link bandwidth.

In order to improve performance, meet the request of application and support quality of service, switch should not only support a large number of priorities, but also have the ability to buffer a large number of packets. The conventional buffering queue architecture can't meet the above requirements well. So we need to design a new priority buffering queue architecture.

2.1 PVOQ BASED ON VOQ

The scheme based on VOQ can be easily scaled to support priority, which is priority virtual output queue (PVOQ), by further grouping the VOQ into subqueues according to their priority. For example, figure 3 shows the queue architecture at one of input ports.

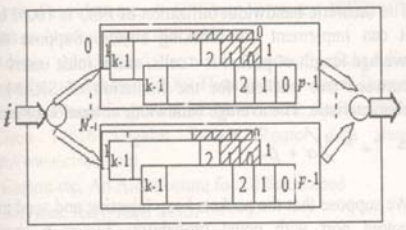


Figure 3 PVOQ mechanism

Each VOQ is divided into p sub queues, p standing for the number of priorities, which the switch supports. Priority 0 is the highest priority, and the lowest is p-1. Assume $Q_{i,j}$ is the queue that buffers cells coming from i input port and going to j output port. Let $Q_{i,j,k}$ denotes the k priority sub queue of the $Q_{i,j}$ ($0 \le k \le p-1$, $0 \le i,j \le N-1$).

As for the switch, because the queue is divided according to priority, there is no HOL among the queues with the same output port and different priority. But there is HOL among the queues with same priority and different output port. Moreover, we know from section 1.2 that with the increasing of the number of port and priority, it's implementation cost will increase a lot.

2.2 PFQ BASED ON FIFO

Section 1.1 has shown that FIFO may cause the HOL. Some researchers proposed that using window mechanism to improve the performance of FIFO. Suppose that the window size is l and l is smaller than the length of queue L. Dispatcher is able to schedule the prior l cells simultaneously, thus decreases HOL. But HOL is completely solved only if l is as large as queue length L. Using window scheme to implement FIFO queue is complex and the benefit is very

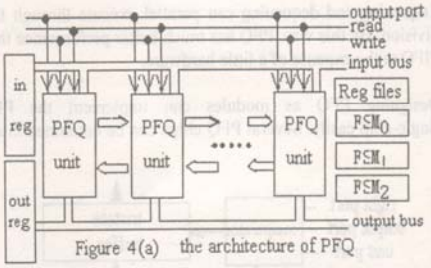


Figure 4(a) the architecture of PFQ

limited. So we provide a priority FIFO queue (PFQ). It uses the idea of shift register, organizing every FIFO queue as a linked list. High-speed local bus is used to solve HOL efficiently. It's architecture is shown in figure 4.

Tags are often set in cells, for instance, output number, sequence number and priority. These bits can be used as control bits. When new cell comes, the principle that decides where it should be stored is shown as the following. First is looking at the output port number of the cell. Cells with the same output port number will be grouped together. Second is inserting the cell into one group according to its priority. The cell with higher priority is arranged at the right of the lower priority cells. If both the output port number and the priority are same, then abide the first in first out principle, the later coming cells are arranged at the left of the former cells.

PFQ module is composed of a series of units, which is similar to a linked list. Each unit buffers one cell. Every time a new cell comes, it will result in a series of shift operation. All lower priority cells buffered will shift to left in order; the new arriving cell will be inserted into proper position. The discard policy we use is that the lowest priority packet will be lost.

At every dequeue operation; it is responsible for every unit to judge if it has the highest priority of the specific output port. If the cell has, then the unit will drive the output bus, export the cell buffered in it and trigger a series of right shift

operation.

The constitution of each unit is not complex. It includes dual port SRAM, tristate buffer, comparator, multiplextor and some essential control logic devices. Dual port SRAM buffers cells. Comparator implements priority compare. Tristate buffer is used to drive an output bus when the local logic decides it has the highest-priority for the requested output port number. Only the one has the highest-priority will dequeue at each scheduling. Multiplextor is used to implement shift operation of queues. This organization has an advantage that control logic is distributed to every unit and every unit make it's own decisions based on its local information. So it is no need for centric control and thus it is easy to scale.

According to the report, nearly 50% message distribution on Internet has the length shorter than 64 bytes,and in order to adapt with other commercial protocol chips, the interior data width of PFQ is 64 bits and the unit buffer size is 64 bytes.

The control logic in PFQ is mainly composed of three state machines, FSM_0, $FSM_1$and FSM_2.

FSM_0: receive cell from input port and enqueue it.
FSM_1: shift cell.
FSM_2: dequeue the cell when the output port is free.

Enqueuing and dequeuing can parallel execute through this division. By this way, PFQ has much better performance than FIFO at the expense of a little hardware.

Designing PFQ as modules can implement the PFQ single-chip easily. Several PFQ chips can be combined into a

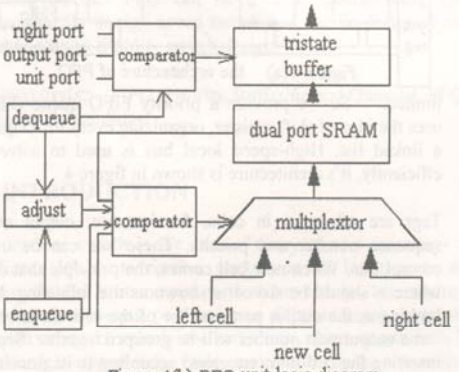


Figure 4(b) PFQ unit logic diagram

large PFQ buffer to store more cells. It is very simple to constitute a switch using PFQ chip. The only thing needs to do is to integrate PFQ chips, crossbar chips and other essential control logic devices.

3. PERFORMANCE ANALYSIS

We use four metrics to evaluate and compare the performance of buffering queues, namely, buffer size, throughput, delay and bandwidth utilization.

Buffer size

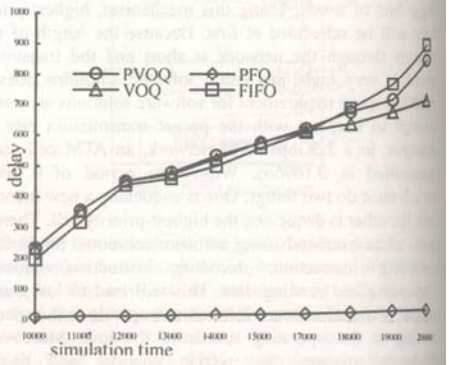
Because SRAM occupies a larger amount of the silicon area, buffer size decides its hardware cost. It is Supposed both the number of input port and output port is N, length of every queue is L, size of every unit is w and the number of priority is p. Table 1 shows the buffer sizes needed by the four mechanisms. We can learn from table 1 that the buffer size PFQ needed is twice as large as FIFO. It is 2/N of VOQ's and 2/NP of PVOQ's. Though it is larger than that of FIFO, it is smaller than both of the other's. Moreover, it avoids HOL completely and supports priority.

Table 1 Buffer size compare

Queue architecture	Buffer size needed
FIFO	$N \times L \times W$
VOQ	$N \times N \times L \times W$
PVOQ	$N \times N \times L \times P \times W$
PFQ	$N \times L \times 2W$

Throughput

Throughput is the number of enqueue/dequeue operations per second. Now, FIFO chip can work at 65MHz or even higher. Assume PFQ also work at 65MHz. Because every enqueue operation needs around 8 cycles, the throughput is about 8.13Mops (millions of operations per second) and the



bandwidth is 8.13Mops × 64byte × 8=4.2Gbps. All these show that PFQ can meet the needs of high-speed switch.

Delay

Delay is the time between cell enqueuing and dequeuing, including the time that a cell in the queue waits to be scheduled. Scheduling algorithm decides the waiting time. It is assumed that the scheduling algorithm adopted is same and we don't consider its affect. At 65Mhz frequency, the cycle time is 15ns, the minimum delay of PFQ at each input port is 15ns*(8 + 1) = 0.135 μ s. We use SIM to simulate. The scheduling algorithm adopted is iSLP, which is a widely used unicast scheduling algorithm. Figure 5 shows the simulation results. We can learn from it that PFQ's delay is much lower than others' after simulate a little longer.

Bandwidth utilization

The theoretic bandwidth utilization of PFQ is 100% because it can implement non-blocking switch. Suppose that the average length of packet is α cells, and it must insert β cells between two packets for the limitation of SRAM access characteristic. The average bandwidth utilization of input port is $\eta_{in} \approx \alpha / (\alpha + \beta)$.

We suppose that the packets keep injecting and send to target output port with equal probability. Figure 6 shows the

simulation results. From it we can learn that bandwidth utilization of PFQ keeps above 0.90 and that of VOQ is stabilize about 0.80, while that of FIFO and PVOQ will

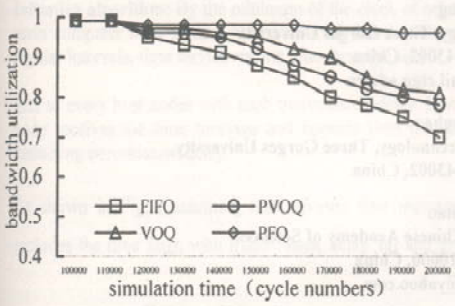


Figure 6 Bandwidth utilization compare

rapidly decline below 0.8

4. CONCLUSION

In this paper, we propose buffering queue architecture PFQ that supports priority QoS. It uses hardware to implement priority buffering and queuing. Although we only discuss the architecture of input queue using PFQ, PFQ can fit output queue too. Through simulation, we illustrate PFQ is flexible, efficient, low hardware cost and can be implemented easily. So it can be fabricated to single-chip to provide module instrument for switch designer. We also designed a new scheduling algorithm PSLP utilizing the benefit brought by PFQ and implemented priority scheduling efficiently.

REFERENCES

[1] S.Keshav and R.Sharma. Issues and trends in router design, IEEE Communication Magazine, 1998 (3)
[2] Henry C.B etc, A Framework for Optimizing the cast and Performance of Next-Generation IP Routers, IEEE Journal on Selected Areas in Communications, 1999 (6)
[3] V. P. Kumar etc, Beyond best effort: Router architectures for the differentiated services of tomorrow's Internet. IEEE Communication magazine, May, 1998
[4] Nick McKeown etc, High Performance Switching (Proposal to Texas Instruments), http://tiny_tera.stanford.edu/~nickm/papers.html
[5] Rajeev Sivaram etc, Implementing Multidestination Worms in Switch Based Parallel Systems, Proceedings of the 24th ACM/IEEE International Symposium On Computer Architure, June 1997.
[6] Nick McKeown etc, Achieving 100% Throughput in an Input-Queued Switch, Proceedings of IEEE Infocom '96, March, 1996
[7] Craig Partridge etc, A 50-Gbps IP router. IEEE/ACM Trans on Networking, 1998 (3)
[8] Adisak Mekkittikul etc, A Practical Scheduling Algorithm to Achieve 100% Throughput in Input-Queued Switches, Proceedings of IEEE Infocom'98, April, 1998
[9] Cisco 12016 Gigabit Switch Router data sheet. http://www.cisco.com
[10] Carson etc, An Architecture for Differentiated
[11] Services, IETF RFC 2475.
[12] H.S. Lee etc, ATM switch with distributed queue
[13] windowing scheme. Electronics Letters 4th , February, 1999 (3)

Study on Time Synchronization of Distributed System

He Peng
Center of Educational Technology, Three Gorges University
Yichang, Hubei 443002, China
Email: hpeng@mail.ctgu.edu.cn

Xia Changhao
Department of Computer Science and Technology, Three Gorges University
Yichang, Hubei 443002, China

Wu Haitao
National Time Service Center, Chinese Academy of Sciences
Xi'an, Shaanxi 710600, China
Email: wht248@yahoo.com

ABSTRACT

Time synchronization is the key foundation of all applications of distributed system. There are two kinds of implement method on time synchronization: the absolute time synchronization and relative time synchronization, the former needs an external time to be as a Primary Reference Source (PRS), the latter can be realized only by executing synchronization algorithm inside the distributed system. In this paper, some time synchronization algorithms used in LAN were discussed in detail, and some application results with some algorithm were also analyzed.

Keywords: Time Synchronization, Distributed System, Time Server, Synchronization algorithm

1. INTRODUCTION

Time synchronization is essential to a computer application system, there is no possibility of dual implications for the time of centralized system [1]. However, there has no standard time unification system or common time base for distributed system. So it is necessary to establish time service system or Time Server for distributed systems to realize time synchronization [2]. At present, there are many studies of multi-media message synchronization transfer, but the achievements on synchronization as system time base are rarely found. Base time synchronization is the base of all application systems, including Supervisory Control And Data Acquisition (SCADA) for power station, emission and tracking system of missile or satellite, Synchronous Digital Hierarchy (SDH) application and network time service of Internet or Intranet etc. All systems mentioned above need time unification with high precision, so system base time synchronization has been used extensively.

This paper first describes a simplified model for timing system and some essential concepts on time synchronization in detail, then focuses on algorithm of time synchronization, and finally presents the results on application of centralized initiative algorithm to SCADA of Gezhouba dam power station.

2. TIMING MODEL

Assume t is physical time in nature and $T(t)$ is time output of any timing system, the simplified model for the time output of the system are listed as follows:

$$T(t) = T_0 + (1 + \beta)t \quad (1)$$

where T_0 is the initial value when physical time t start timing, and $(1+\beta)$ is increasing coefficient:

$$\frac{dT(t)}{dt} = 1 + \beta \quad (2)$$

Evidently, when $\frac{dT(t)}{dt} = 1$, i.e. $\beta = 0$, it shows that the timing system is wholly synchronous with natural increasing physical time, which is a ideal status. However generally $\beta \neq 0$, and there is always an increasing bias coefficient between the timing system and physical time, which has relation to the physic feature of the timing device and is generally described with maximum draft ratio of the device.

Therefore, the time synchronization refers to having a maximum β for any timing system, and has the following relationship:

$$1 - |\beta| \le \frac{dT(t)}{dt} \le 1 + |\beta| \quad (3)$$

and within the fixed time Δt, maximum clock bias $2\Delta t\beta$ against physic time is not greater than δ. To hold the synchronization relationship, the clock frequency should be brushed periodically, be synchronized with standard time, or be calibrated at regular intervals. The brushing period is $\varepsilon = \frac{\delta}{2\beta}$, and the brushing frequency is $f = \frac{1}{\varepsilon}$.

With a worldwide time service, Universal Time Cooperated (UTC) is usually used to record the physic time t, so the timing systems of all other application systems should be synchronous with UTC.

3. TIME SYNCHRONIZATION ALGORITHM

Centralized algorithm

The algorithm refers to using a host node in distributed system as a time service, which has time base with high stability or has measures to synchronize with external time

base such as UTC. Other host node realizes time unification with Time Server by executing algorithm includes initiative algorithm and passive algorithm.

Initiative algorithm: By the minimum of the clock of other main computer node against to the clock of time service at regular intervals, time service reports time message, which gets to every host nodes with each transmission delay. Host node receives the time message and corrects time though deducting transmission delay.

As shown in Fig.1, assuming that reported time message includes the time *Ts(t)*, with transmission delay t_{Ri} and get

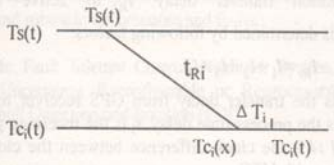


Fig.1.Direct broadcasting time

to node *i*, therefore corrected time is:

$$Tc_i(x) = Ts(t) + t_{R_i} \tag{4}$$

The clock bias against *i* node clock is:

$$\Delta T_i = Tc_i(t) - Tc_i(x) \tag{5}$$

i node can correct the clock by ΔT_i. Evidently, correcting error is mainly decided by the measure error of t_{Ri}. With any node, t_{Ri} always consist of a minimum transmission delay and a random network congests delay, . i.e. $t_{R_i} = t_{\min i} + t_a$. By accumulating and statistic treatment, a mean t_a can be computed for t_{Ri} to be more accurate.

The above algorithm is fit to the status in which Time Server has synchronization measure with external time base. If having no external time base, the whole system only depends on the base of Time Server to unification the time. An improved method is Berkeley algorithm [3]. The principle is as follows: Time Server send inquiring message to all N nodes periodically, and all nodes pass back the time message according to the inquire after having received the all time information, Time Server computes a mean, and then sends out these means.

As shown in Fig.2, the algorithm of getting the mean and the correcting of host node are as follows:

$$TSi(t) = TCi(t) + t_{Ai} \tag{6}$$

$$\overline{TS}(t) = \frac{1}{N}\sum_{i=1}^{N} TSi(t) \tag{7}$$

$$TCi(x) = \overline{TS}(t) + t_{Ri} \tag{8}$$

$$\Delta Ti = TCi(t) - TCi(x) \tag{9}$$

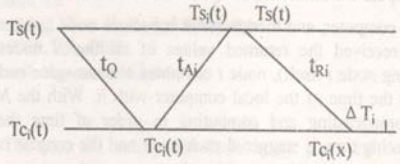


Fig.2 Broadcasting time Series of mean value

The sending time message is a statistic value, so the algorithm don't need accurate physic time base, but need computing process to have high resolving power.

Passive algorithm: Initiative algorithm needs time server to broadcast and inquire periodically, don't distinguish the brushing periods of every nodes. In passive algorithm, according to the require of resolving power of task and the brushing period relative to time server, each node decides if to inquire the time server in order to gain time synchronization. In this case the Time Server is passive,

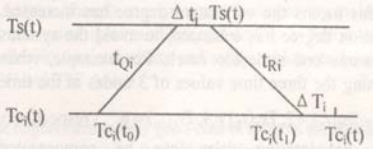


Fig.3. Synchronization time series asking for service

that's called Cristian algorithm [4].

As shown in Fig.3, node *i* sends inquire message at time $Tc_i(t_0)$, and it gets to Time Server after t_{Qi}. The responding time Δt_i means that after Δt_i Time Server send time message. After transmission delay t_{Ri}, node *i* gains the corrected time $Tc_i(t_1)$, therefore:

$$Tc_i(t_1) = Ts(t) + \Delta t_i + t_{Ri} \tag{10}$$

For the sake of convenience of measure, for *i* node, assume the round time is T_{Round}.

$$T_{Round} = Tc_i(t_1) - Tc_i(t_0) = t_{Qi} + \Delta t_i + t_{Ri} \tag{11}$$

When round transmission time is equal i.e. $t_{Qi} = t_{Ri}$, therefore:

$$TCi(t_1) = TS(t) + (T_{Round} + \Delta t_i)/2 \tag{12}$$

$$\Delta T_i = TCi(t) - TCi(t_1) \tag{13}$$

Since independent of transmission route, the measuring and evaluating of T_{Round} or Δt_i can be made by local computer, compute the ΔT_i and realize synchronization.

Distributed algorithm

Evidently, centralized algorithm relies heavily on Time Server, which is the distinct defect. To improve it we can use distributed algorithm. In the network with *M* host nodes, we define *N* nodes as Time Servers. In order to decease the transmission amount of network message and make the

algorithm practical, the value of N should be medium. If node i (the any node of N nodes) at a time broadcasts its time to other N-1 nodes, any other nodes will receive the time, and compute the difference $\Delta Ts_{ij}(t)$ between it and the time of local computer, and then transmit it back to node i. After having received the returned values of all the N nodes (including node i itself), node i computes a mean value and corrects the time of the local computer with it. With the N nodes broadcasting and computing in order of time the broadcasting time is staggered each other and the congest is avoided

$$\Delta\overline{T}s_i(t) = \frac{1}{N}\sum_{j=1}^{N}\Delta Ts_{ij}(t) \tag{14}$$

$$\Delta T_i = Ts_i(t) - \Delta\overline{T}s_i(t) \tag{15}$$

The algorithm has the following features: 1) The broadcasting progress is the same as the centralized initiative algorithm and the error in computing $\Delta Ts_{ij}(t)$ is mainly decided by transfer delay. 2) The returned value $\Delta Ts_{ij}(t)$ of node i is a difference value, so it has no relation to return delay. 3) Within the period for all N nodes to finish the computing, the correlative time of any node to another is always 2, this means the correlative degree has increased, and distribution degree has enhanced, to avoid the systems relying on some one node too much. For example, when N=3, assuming the three time values of 3 nodes at the time t_0 are: $Ts_1(t_0)=2, Ts_2(t_0)=4, Ts_3(t_0)=6$, respectively considering broadcasting delay can be compensated accurately, can be calculated out $\Delta\overline{T}s_1(t_0)=2, \Delta\overline{T}s_2(t_0)=0, \Delta\overline{T}s_3(t_0)=-2$. The time of the 3 nodes can be synchronized at $Ts_i(t)=4(t>t_0)$ with these values.

To make further improvement of the algorithm mentioned above, at first, we reject the maximum and the minimum of N $\Delta Ts_{ij}(t)$ values, secondly compute the mean value to avoid all the computing of $\Delta\overline{T}s_i(t)$ losing the significance with too much error due to delay jitter.

4. APPLICATION AND CONCLUTION

Centralized algorithm needs server or main node has a time base with high stability, or has a measure that can synchronize with external PRC. For the former (passive server), the synchronization relation between other nodes and Time Server or main node is a quasi-synchronous. For the latter (active server), it needs to depend on some kind of synchronization measure to realize active synchronization, the mainly measures are GPS, GLONASS, LORAN-C and BPL, BPM in China.

Distributed algorithm is purely a statistic algorithm, which has lower demand for at time base and has higher demand for accuracy and efficiency of the algorithm. Since the synchronization time computed using this algorithm reflects the time feature of the network well, we can describe this feature by "network frequency".

We used the time synchronization algorithm mentioned above in this paper in the SCADA of Gezhouba dam power station. The distributed SCADA has the structure of double servers and double bus backups, in which GPS receiver outputs time synchronization signal which one of is the GPS data flow to be sent to time server directly, another of which is the 1pps pulse signal with the width of *100ms* to be sent directly to synchronization device of control unit. The synchronization error of the two time signals with UTC is less than $1\ \mu s$. The detail realizing progress is also presented in reference [5].

In order to decrease indeterminate factors in time transfer to the greatest extent, we use initiative algorithm. Considering the possible line break of GPS receiver due to some reasons such as breaking out of sun' spots, we adopt the method combining the active initiate algorithm with passive initiate algorithm to guarantee against the break of time synchronization in all the SCADA system. The time synchronization transfer delay t_{Ri} in active initiate algorithm is determined by following factors:

$$t_{Ri} = t_1 + t_2 + t_3 + t_4 \tag{16}$$

Where t_1 is the transfer delay from GPS receiver to Time Server, t_2 is the process time delay, t_3 is the receiving process delay, *and t4* is the clock difference between the clocks of Time Server with UTC.

In the practical engineering, estimating the errors on the above links, we gained synchronous total error of active initiate algorithm as follows:

$$\begin{aligned} E(t) &= \sqrt{\Delta t_1^{\,2} + \Delta t_2^{\,2} + \Delta t_3^{\,2} + \Delta t_4^{\,2}} \\ &= \sqrt{100^2 + 100^2 + 200^2 + 720^2} \\ &= 760.53us < 1ms \end{aligned} \tag{17}$$

With respect to passive initiate algorithm, there is no t_1 link, the total error is less than *E(t)*. Comparing with passive algorithm, active algorithm can make the time synchronization error of all the SCADA system controlled within *1ms*. For passive algorithm, though its error is smaller in a short time, but with time going on, the relative time of all the SCADA system to UTC will be more and more. So passive algorithm could be a complement when active algorithm is invalid (line break). Combining the active algorithm with passive algorithm can guarantee a better accuracy of the time synchronization of all the SCADA system.

REFERENCES

[1] H. Kopetz, G. Grunsteidl, "Clock Synchronization in Distributed Real-Time System", IEEE Trans. on Computers, Vol.36, No8, 1987, pp.933-940

[2] B. Liskov, "Practical Uses of Synchronized Clocks in Distributed System", Distributed Computing, 1993(6), pp.211-219

[3] F. Lamport, "Time, Clocks and Ordering of Events in a Distributed System", Comm. of the ACM, 1990,21(7), pp.558-564.

[4] F. Cristian, "Probabilistic Clock Synchronization", Distributed Computing, 1993, (6): pp.146-158.

[5] He Peng, "Implementation of Time Synchronization of DCS with GPS", Journal of Three Gorges University, 2001, (1), pp.44-47.

Reviews of Fault Tolerant Control for Nonlinear System*

Huaping Shao and Jiashu Xu
The School of Electronic & Information Engineering, Xi'an Jiaotong University
Xi'an City, Shanxi Province, 710000, P.R. China
Email: togorun@263.net and E-mail: togorun@163.net

ABSTRACT

This paper presents the general principle and application of fault tolerant control for complicated nonlinear system. A new scheme, analyzing and explaining the truth of fault tolerant control, is put forward. The presented control scheme can effectively identify and accommodate nonlinear unknown faults, and the controlled system is stable and robust in uncertainties and faults.

Keywords: Fault Tolerant Control, Nonlinear System, Complex System, Uncertainty, Reconfigurable or Restructurable Control (REC), Fault Detection and Isolation (FDI)

1. INTRODUCTION

Complexity Science is a 21st century's science. Complex system are characterized by poor models, high dimensionality of the decision space, distributed sensors and actuators and decision makers high noise levels, multiple subsystems and performance criteria, complex information patterns and overwhelming amount of data and stringent performance requirements.

The difficulties in the control of complex systems can be broadly classified into three categories.

- Nonlinearities
- Complexity in plant controller and environments
- Uncertainty

Increasing requirement on productivity, function and performance lead to plant and controller more and more complex, and operating near design limits for much of the time. This may often result in system faults or failures. Modem controlled system has demanded performance requirements under a variety environment.

In complex system, there may be a number of configurations. Configuration is the plant in each fault mode or dynamic change. A configuration is a behavioral space of extended by the specification of all system parameters, model and constraints on the applicability of particular models. Controlling such a complex system need use reconfigurable or restructurable controller. There are two kinds of restructurability, one is spatial restructurability which involves changes of model within the same configuration, as in multiple subsystems. Another type is temporal restructurability that results form the variation of the system structure over time, as in change due to faults.

Reconfigurable or restructruable control (REC) is a relatively new scheme in the design and development of control systems. In fault conditions, REC is also called Fault Tolerant control (FTC). There are two direct needs driving its development;

- The need for controlling plants that at change their dynamics structurally in an unpredictable fashion, meaning that at different points in time, the dynamic model of the plant has to be descried by equations having different variables and different mathematical operators.
- The need for dealing with faults in the plant and controller, meaning that a reconfigurable controller can change its parameters as well as its structure in order to compensate for a structural change (e.g. a fault) in the plant or the control system itself.

Intelligent control (IC) is the ability of control system to operate successfully in a wide variety of situations by detecting the specific situation that exists at any instant and appropriate servicing .IC has the ability of adaptation, learning, pattern recognition, decision making and self-organization. A controller responds with speed and accuracy to sudden and large changes may be considered as having intelligence. REC or FTC is key part of IC system.

In addition, there is an urgent need for improving the dependability of automated systems. Dependability is a fusion of reliability, availability, maintainability and safety, which can be enhanced by REC and FTC design.

The requirements of any good control systems are stability, accuracy and speed. Achieving these in complex systems, in the presence of large uncertainty, nonlinearities and dynamic changes, is the challenge for control theories today.

Conventional controllers do not possess all the attributes in a wide variety of situations. Control system design has traditionally been based on single fixed or slowly adapting model of the system. This implicitly assumes that the operating environment is either time-invariant or varies slowly with time. In complex system, it is usually not true. Many types of changes other than slow parameter variations are encountered, e.g. faults in the systems, failures in sensors and actuators, external disturbances, changes in subsystem and in the system parameters. In general, complex systems operate in multiple environments that may change abruptly from one configuration to another.

Changes in environments, plants, controllers and performance criteria, unmeasurable disturbances and component or system faults are some of the features which necessitate fault tolerant control, reconfigurable control as well intelligent control. Form above review, it can be seen that REC and FTC is a very important issue in the design of modem system.

2. METHODOLOGY FOR FTC

There are several control schemes that are used to deal with the issues of structural changes and faults:

- feedback control
- gain scheduling
- adaptive control
- robust control
- switching control
- expert control
- learning control
- autonomous control
- intelligent control

One efficient REC or FTC system should be a hybrid system that integrates and fuses above control algorithms. In general, these

schemes can be classified as two ways. One is called passive fault tolerant control (PFTC) scheme which uses such as adaptive controller or robust controller with feedback control to accommodate faults. Its main objectives are to achieve the stability and integrity of the system. Another is called active fault tolerant control (AFTC) scheme that reconfigures or restructures controllers or control laws based on fault detection and isolation (FDI) and reconfigurable logic or switching logic. Its goals are to retain system stability as well as improve performance. AFTC is able to enhance system dependability roundly. This is a development trend of FTC.

3. THE SOLUTION TO FTC

A possible solution to AFTC is showed in fig.1.

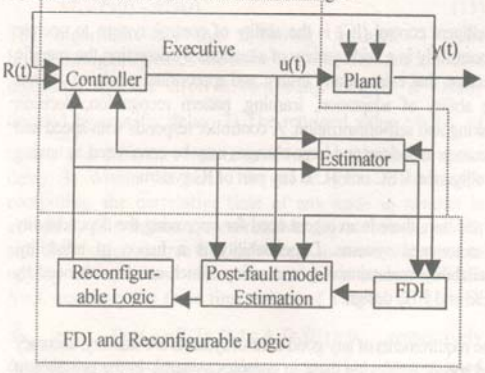


Fig1: A scheme for AFTC

Five main tasks:

- Executive control
- Fault detection
- Fault identification
- Fault mode estimation
- Reconfiguration logic

4. COMMENTS

- Redundancy is necessary design of FTC system. Parallel or analytical one can provide redundancy. Parallel redundancy is the duplication of the controller or system hardware.

- Feedback controller typically reduces the plant' output sensitivity to measurement errors and disturbance inputs, so it provides certain closed-loop stability as well, if the plant is lightly damped or unstable.

- Reconfigurable means that control system' structure or parameters can be altered in response to system faults. Reconfiguration is based on the control system detects, identifies, and isolates faults, and modifies control laws to maintain acceptable performance. FTC system through configuration is both adaptive and redundant, and also is intelligent system. Reconfiguration attempts to retain stability and performance characteristics.

- Double threshold is need in FDI and reconfiguration logic. Note: not all faults need control reconfiguration. Only when the system' stability and performance are not accepted, so it need two threshold. One is as for FDI, another is for reconfiguration logic. In general, the threshold value for FDI is smaller than that of reconfiguration.

The methodologies for accommodating anticipated faults, which the post-fault system characteristics are known a priori, and unanticipated faults, which are not directly recognized by FDI, and exists a tradeoff between the time (speed) and accuracy to attain a solution to the reconfiguration problems and generality of the approach..

Nonlinear FDI scheme and FTC is a challenge in design of FTC system. The conventional FDIA and FTC are for linear system subject to additive faults, based on linear modeling or estimation techniques. Feedback linearization is to transform the nonlinear system into a equivalent linear one through a change of coordinates and nonlinear feedback..

Neural Networks(NN) are suited to cope with these categories of difficulties, and have been shown them extremely efficient as pattern recognizes (for FDI), and have also been proven to be very efficient for the identification and control of nonlinear dynamic systems through a combination of both off-line and on-line learning.

Fuzzy Logic systems (FLS): FLS is a name for the systems, which have a direct relation with fuzzy sets, linguistic variables, and so on.

5. PRESENT SITUATION

For anticipated faults, additive faults in linear system, a AFTC scheme based on condition monitoring and fault diagnosis (CMFD), and State observer and switching logic (multiple models) has been presented. For unanticipated faults and nonlinear system, it has not yet presented an efficient method (scheme).

In the case of unanticipated faults, learning methodologies are required to perform simultaneous on-line identification and control of the post-fault dynamics. This corresponds to indirect adaptive control, which is well known in the adaptive linear control. In non-linear case, however, the problem becomes considerably more complex because the control is required to reject of the fault by canceling the nonlinear function representing the deviation in the dynamics due to a fault.

The objective of a learning scheme is to develop an adaptive procedure that not only detects changes in the dynamics but is also able to learn (i. e. create a rough mode) these changes for the purpose of indenting and correcting the fault. Learning is an inherent component of FDIA and AFTC architecture for unanticipated faults

REFERRENCES

[1] S. A. Reveliotis, M. M. Kolar, A framework for on-line learning of plant models and control policies for reconfigurable control, IEEE Trans. on Systems, Man, and Cybernetics, 1997, Vol.25, No.11, pp1502-1512.

[2] K. S. Narendra, S. Mukhopadhyay, Intelligent control using neural networks, IEEE Control Systems, April, 1992, pp11-18.

[3] Anthony. J. Calise, nonlinear flight control using neural networks, 1997, J. of Guidance, Control, and Dynamics, Vol20, No.1, pp26-33.

Identifying Document Dependency on Web Server

Weiping Zhu
School of Computer Science
The University of New South Wales, ADFA
Australia ACT2600
E-mail weiping@cs.adfa.edu.au

ABSTRACT

The WWW has become the dominant tool in the Internet, which takes 75% network traffic and the trend is continuing. To improve system performance in terms of response time, cache has been widely used. Study shows traditional cache methods, e.g., proxy and browser, can only marginally improve system performance. Hence, server assisted caching has been proposed which can substantially improve system performance. However, server assisted caching depends on the knowledge of document dependency, how to obtain it from incomplete information has not been thorough studied. In this paper, we propose a method that uses the maximum likelihood method to identify the correlation between web documents. Preliminary study confirms that the correlation identified by the method can improve system performance.

Keywords: Web Caching, Performance Evaluation.

1. INTRODUCTION

The success of the World Wide Web (WWW) has brought an exponential growth of the user population, which creates heavy traffic on the Internet. Statistics shows that over 75% of network traffic is web related and this trend is increasing [1]. Due to the continuous increase of traffic, end users from time to time experience long delays or denial of service when they visit some web sites. All of these, on one hand, shows that the WWW has become the major source for people to distribute and/or obtain various information; on the other hand, it shows the enormous network traffic created by Web surfing starts to threaten the services provided by the WWW and affect network performance.

Although the advance and deployment offabrics give the impression that scaling the Internet is simply an issue of adding more resources, the Internet's growth exposed this impression as amyth [2] since information access has never been, nor it is likely to be in the future, evenly distributed. Flash-crowds, i.e., millions of users hitting the latest hot site at once, are very common that can overload some servers and subsequently create network congestion [3]. Once this happen, requests sent to these servers can suffer from long delay, and even lead to denial of service.

Past experience tells us to resolve the threat; caching and replication are the only effective methods to meet the exponential growth and uneven distributed user demands. Caching transfers remote accesses to local ones that can substantially reduce network traffic and provides highly demanded speedup; replication distributes incoming requests to a number of servers based on proximity. In this paper, we propose a server assisted caching method which is different from the previous ones [4, 5, 6] in its dynamic feature, i.e., the method, based on a Maximum Likelihood Es-timator (MLE), can identify the correlation between documents in the y. This feature allows it to be used in those web sites in which documents and their correlation vary over time, such as news, stock. The MLE has been used in our preliminary study in which a 30 days server log was used to simulate two environments; one can learn from history, the other cannot. Simulation shows the proposed method can extract common behavior on the y and subsequently increase the hit ratio by 10%.

This paper is organized as follows; we discuss the related work in web caching in the next section. Section 3 is used to present the model used in the proposed method. Simulation is carried out on the data provided by our department web server. The last section is devoted to conclusion.

2. RELATED WORKS

Client caching, including browser and proxy caching, has been widely used in today's WWW, which requires documents to be fetched to the client's side first, then temporal-logic is used to speculate which documents need to be cached or prefetched. This technique stems from operating systems in which very recently accessed pages are kept in memory since these pages have high probability to be accessed again in the near future. The effectiveness of this method has been demonstrated in many operating systems and distributed systems because of the intrinsic temporal locality within a process. However, research shows using the same technique on the WWW can only lift the hit ratio to 30% - 40% when there is unlimited cache space, otherwise it could be as low as 6% [7]. The inability to achieve better performance by using temporal logic in the WWW is because web requests do not have the same temporal locality as processes, i.e., a user would not repeatly view the same web page. In fact, recent study shows that the temporal locality of web requests may be proportional to the popularity of documents since the distribution of interarrival time is hardly changed after a random permutation of the logged requests in a server log[8].

The limitation of browser and proxy caching triggers enormous interesting in searching for new solutions, one of them is server assisted caching [9, 6, 5]. Server assisted caching differs to its predecessor in which servers, not only clients, play an active role to speculate documents to be cached.

In order to correctly speculate the documents that users may access, a server needs to know document dependency that represents casual relations between documents on the basis of visiting probability. Provided with the probabilities, when a server receives a request from a client (or its proxy server), apart from the document, the server sends those documents that are most likely to be requested subsequently to the client or its proxy server. By this means, when the client wants to

visit next document, the required document has a much higher probability than other methods to be cached locally. Then, access latency and network bandwidth are reduced. Preliminary study shows that with the knowledge of dependency, server assisted caching can substantially improve system performance in terms of access latency and hit ratio [9].

In [9], Bestavros studied the server assisted caching and concluded the this method can efficiently reduce both server load and service time that is above and beyond what is currently achievable using client-based caching. However, in order to obtain document dependency access logs collected by a web server (cs-www.bu.edu) for a month was thoroughly analyzed to compute the document dependency, which was conducted of ine because of time-consuming. Therefore, the method is not suitable for those web sites in which contents and reference links are changed over time; we call them time-sensitive web sites in the rest of the paper.

Yan et al studied the possibility of identifying user access patterns from log files [10]. They, instead of supporting caching, aims to dynamically identify users' interests and create web pages according to interesting. In order words, a web server can provide personalized web pages to assist users' surfing.

3. MODEL AND METHOD

Server assisted caching depends heavily on the knowledge of document dependency. If server log recorded all accesses, document dependency would be obtained by computing access frequency. However, due to various caches located between a client and servers, a server in practice can only observe the requests that cannot be served by caching. Under this condition to obtain document dependency from a server log, new methods are needed which can estimate the information missed from the server log.

3.1 Logical Structure

A user surfing a web site normally starts from a web page, then based on the references embedded in the page to access other documents. This process is conducted in a recursive manner although a user may backtrack sometimes. Thus, knowing the structure of the documents kept in a server can assist us to find the missing information, and finally identify document dependency.

Despite circular references may exist in a web site, all documents stored in a web server logically form a hierarchical structure as shown in Fig 1.

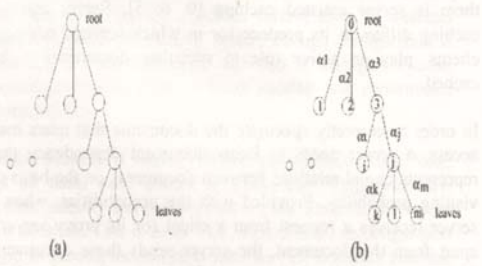


Figure 1: Structure

(a). The root HTML (or its like) file has references to its children that can be HTML, JPEG, GIF, etc. Each of these documents as its parent has its references to its children, this process continues until it reaches to pure documents that have no references to other objects.

A tree or trees can represent the logical structure of the documents stored in a web site.Let T = (V,L) denote a tree of V nodes and L weighted links, as shown in Fig1 (b). A node corresponds to a document; the child nodes of a node are the reference documents embedded into the parent document. A link between two nodes shows the existence of dependency between them.

The degree of the dependency is represented by a conditional probability, P[child|parent], which denotes the access probability of the child node under the condition of its parent shaving been accessed. The value of the probability is used as the weight linking the parent to the child node. All children of node j, (j $\in$V), form a set c(j). Each node, apart from the root, has a parent, function f(k) returns node k's parent, e.g.j= f(k) such that < k,j>$\in$ L. The dependency between the two nodes can be written as $k \rightarrow j$.

3.2 Data Model

To find document dependency, requests received by a server need to be analyzed by a statistical method. A Bernoulli model is used in analysis, which means there are only two outcomes for a sampling point during an experiment - a) accessed or b) not accessed. If accessed, the access is highly likely following the path from its parent. If not,

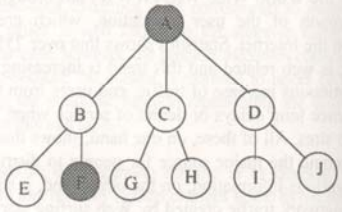


Figure 2: Inference Example

its parent may or may not be accessed. Thus, a request down to a document is described by a stochastic process X = (X_k), k $\in$V as follows;

X_k = 1 if document k is requested, otherwise X_k = 0. If X_k = 0, then, X_j = 0; $\forall j \in c(k)$ .Otherwise, $P[X_j = 1|X_k = 1] = \alpha_j$ and $P[X_j = 0|X_k = 1] = 1 - \alpha_j$. Here, α_j represents the dependency of document j on its parent, document k. As a probability, $0 < \alpha_i < 1$; $\forall k \in V$. Finally, let $\alpha = (\alpha_i)$, i $\in$V . Here, the task is to identify α from collected samples.

As previously stated, a server can only observe a part of the requests sent by its users. To identify α from the incomplete information, inference must be applied. Let $\rightarrow$ donate the access-before relation between two consecutive accesses. For instance, a server may receive two requests one after the other for the two shaded documents shown in figure 2 and A$\rightarrow$F. The server can derive a conclusion that the user is very likely reach F, via B. If samples show the access frequency of A$\rightarrow$F

is much higher than A→X,X∈{E, G, H, I, J},the server has very good reason to believe that ABF are closely related. Next time, if A is requested by a client, apart from A→F is sent to the client.

This example shows the principle of our approach. In order to be more accurate to group related documents, we need to find out the degree of dependency between documents. Statistical inference is used on the collected samples to estimate α. Once the dependency is identified, it will be used to group documents that assists various caching servers to get the documents in advance. The objective is to increase hit ratio and reduce access latency and network traffic.

3.3 Method

To overcome ash-crowd phenomenon, in particular for time-sensitive web site, a server should be able to identify document dependency in a timely and accurate manner because the dependency will be used as basis of clustering documents into groups. Later, if a document is required, other related documents in the same group are sent to the client. In order to adapt a dynamic environment, the proposed method only monitors the requests sent to a number of selected documents. Then, a Maximum Likelihood Estimator (MLE) is applied on the collected samples to infer the dependency of related documents from the incomplete information. In a web site, there is only a part of the documents that are time-sensitive. Instead of monitoring all requests and process them that is time consuming and confusing, some nodes located at sensitive points are selected, the process could divide the tree structure into a number of sub-trees.

For each sub-tree, only the accesses to the roots and leaf nodes are collected. The accesses are further divided into clusters on the individual basis, and also based on a session concept which is defined as a set of requests from a site that falls into a time frame in which the time interval between two consecutive requests does not exceed a given $\triangle$.

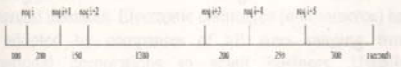


Figure 3: Session Example

Fig. 3 shows an example in which a server observes the requests sent from a client. If $\triangle$ is 1000 seconds, the first three accesses are in a session, while the other three belong to another. Sessions aim to group related requests, these requests are not only temporally related, but also highly likely to be casually related, which are extremely important in the process of identifying document dependency. The samples collected from a session is an n-element vector, $<v_0, v_1, \ldots v_{n-1}>$, where n is the the number of nodes that are selected as sampling points and vi is the outcome of sampling on point i. As previously stated, if a leaf node is accessed and no matter how many times, vi = 1, otherwise vi = 0. Therefore, the sample space collected in a period is the set of all possible configurations, $\Omega=\{0,1\}^n$

For each x^2, let k(x) denote the number of sessions for which the sample x is obtained. The probability of k independent sessions $x_1, x_2, \ldots x_k$ is then $p(x^1,\ldots x^k;\alpha) = \prod_{x\in\Omega} p(x;\alpha)^{k(x)}$

where α is the set of document dependency within the hierarchical structure. Solving the equation, we can obtain α. Maximum Likelihood Estimation (MLE) is used to infer α_j. A threshold, γ, is used by the classifier to determine the degree of dependency between two documents. If $p_{i,j} \succ \gamma$, document i is considered to be closely related to document j. Then, if i is requested, the server sends document j as well.

4. SIMULATION AND RESULT

To measure the effectiveness of the proposed method, two simulations were conducted, one, called old, is based on the traditional browser/proxy cache method, the other called new, apart from having browser caching, adopted the server assisted caching. The Hit ratio is used as the criterion in the simulations to evaluate the performance of these two approaches.

The simulations were carried out in the same condition and driven by a 15-day server log collected at the Department of Computer Science, the University of Queens land, which has more than 50,000 request entries. Based on the log file, the document structure kept in the server was identified and used in the simulations. A client server model is used in the simulation, the client sends requests to the server based on the log file, the server retrieves the requested document and sends it back to the client. The simulation that has server-assisted function works in two phases: warm-up and process. In the warm-up phase, the server does not have any knowledge about the user access patterns; it serves the incoming requests in one by one basis. Meanwhile the server based on the requests received learnt the common access patterns. A simplified MLE was developed and used in the learning process. In the process phase, the server based on what it learnt in the warm-up phase serves the client. Once, a client requests a document that belongs to a common access pattern, the server pushes related documents together to client. Although the simplified method cannot take the advantage of the proposed method fully, it is good enough to show the advantage over the traditional method.

Another question considered here is the length of warm-up period. Obviously, a longer period statistically ensures the identified access patterns are correct, but it may not be able to exploit the discovery early. In addition, long warm-up makes a system suffers from slow reaction to pattern change. In our simulation, two different lengths were adopted, one is 1 hour, the other is 2 hours.The result is shown in table 1.

Table 1: Comparison between the two methods (hit ratio)

Method	1 hour warm-up	2 hours warm-up
Old	0.5395	0.5469
New	0.6539	0.6978

The result clearly shows the MLE based method has better performance than the traditional method. In terms of warm-up period, the result shows the 2 hours warm-up can produce better performance than the 1 hour one. Since the documents and the corresponding structure are no-change during this period, there is no need to repeat the warm-up operation. If documents and/or the corresponding structure kept in a server change frequently, such as news and stock analysis, the

warm-up operation needs to be repeated accordingly. For this reason, a Bayesian based approach is proposed and under investigation, which is able to learn access patterns in a progressive manner that can not only handle the frequent changing environment, but also refine the identified access pattern.

5. CONCLUSION

The proposed method has been implemented and tested on a log file. The result shows the proposed method can increase hit ratio by at least 10%. The success of the initial study encourages us further extends the work into more general environments with few restrictions. This requires us to search for other efficient methods. Our next task is to study methods that can determine document groups. So far, there are a few prior works on this area, and adopted models are almost limited into Markov models [4, 11, 12], which simply predict users' future requests, conditioned on the current request. In fact, there is some evidence that web-surfing behavior may not be Markov in nature [13].

REFERENCES

[1] K. Claffy, G. Miller, and K. Thompson. The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone. In Inet'98, July 1998.

[2] Lixia Zhang, Sally Floyd, and Ven Jacobson. Adaptive web caching. In Project overview, 1999.

[3] Margo Seltzer. The World Wide Web: Issues and Challenges. In Presented at IBM Almaden, http://wwww.eece.harvard.edu/margo/slides/www.html, July1996.

[4] A. Bestavros. Speculative data dissemination and service to reduce server load, network traffic, and service time in distributed information systems. In proc. of the 1996 Conference on Data Engineering, 1996.

[5] J. Gwertzman and M. Seltzer. An Analysis of Geographical Push-Caching. In ICDCS97,1997.

[6] J. Gwertzman and M. Seltzer. The Case for Geographical Push-Caching. In The 1995 Workshop on Hot Operating Systems, 1995.

[7] A. Bestavros. WWW traffic reduction and load balancing through server-based caching. IEEE Concurrency, pages 56{67, 1997.

[8] S. Jin and A. Bestavros. Temporal locality in web request streams. In Technical Report BUCS-TR-1999-014, 1999.

[9] A. Bestavros. Using speculation to reduce server load and service time on the www. In Technical Report BUCS-TR-1995-006, 1995.

[10] T.W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In Fifth International World Wide Web Conference, May 1996.

[11] I. Zuckerman, D. Albrecht, and A. Nicholson. Predicting user's request on the www. In proc. of the 7th International Conference on User Modeling, 1999.

[12] J. Borges and M. Levene. Data mining of user navigationpatterns. In proc. of the 1999 KDD Workshop on Web mining, 1999.

[13] B. Huberman, P. Pirolli, J. Pitkow, and R. Lukose. Strong regularities in World Wide Web Surfing. Science, 1997.

Research and Development of Negotiation Mechanism in E-Commerce

Sun Ning
Department of Computer Science & Engineering
Beijing Institute of Technology
Post Code 100081 P. R. China
Email: nick_sunny@263.net

Cao Yuanda
Department of Computer Science & Engineering
Beijing Institute of Technology
Post Code 100081 P. R. China

ABSTRACT

This paper describes multi-agent based automated negotiation between clients in e-commerce. An automated negotiation model and architecture of the negotiation agent are presented. A constraint satisfaction-processing component is developed to evaluate negotiation proposals against the defined constraints and negotiation strategic rules. A preference-scoring module performs quantitative analysis of alternative negotiation conditions. A prototype of e-commerce system is implemented to demonstrate automated negotiations among buyers and suppliers.

Keywords: E-commerce, Distributed Computing System, Multi-Agent System, Automated Negotiation.

1. INTRODUCTION

In recent years, distributed computing applications are being increasingly used in a wide range of industrial and commercial domains. Electronic commerce (e-commerce) has been adopted by companies of all sizes ranging from multinational corporations to small business. Usually, individual consumers and companies often want to negotiate the price, the delivery date, the quality of goods and services, as well as other purchase conditions. It is desirable to carry out this negotiation process automatically in e-commerce. Negotiation in e-commerce is a process of exchanging proposals and counter-proposals between two or more parties until a mutual agreement or disagreement is reached.

In this paper, we present multi-agent based automated negotiation in e-commerce that is core issue in electronic business transactions. Our research focuses on bilateral and multi-lateral multi-issues bargaining, which is very challenging research in negotiation.

The remainder of this paper is structured as follows. An automated negotiation model is described in section 2. Architecture of negotiation agent is presented in section 3. Section 4 describes the negotiation process and the proposal Evaluation model in more detail and section 5 concludes the paper.

2. AUTOMATED NEGOTIATION MODEL

The model is mainly made up of four parts: negotiation agent, Web servers, client computers, and application systems as shown in figure 1.

In the model of automated negotiation, sellers can publish information about the goods and services they provide on their home pages, which can be browsed by the buyers using Web browsers and search engines. Through searching and browsing, a buyer can identify potential sellers with whom the buyer wants to conduct negotiations. Negotiation agent and Web server, which are integrated together, carry out negotiation on behalf of clients on Internet by information of goods and service provided by clients.

In addition, sellers have application systems to maintain the inventories of goods or the information about the services they provide. To simplify our description of the negotiation process, we use the symbols BU and BNA to denote the user and negotiation agent of party buyer, respectively. The corresponding symbols SU and SNA are used to denote those of negotiation party seller.

BU initiates the negotiation process by issuing a request for a product or service to his negotiation agent. BNA carries out the negotiation process by sending a CFP (call-for-proposal) to SNA, which represents the seller SU. SNA checks the information and constraints specified in the CFP against the defined information of SU. If constraint violation is found, relevant strategic rules, which have been provided by SU, are applied to either (1) reject the CFP, (2) accept CFP, or (3) make modification to SNA, or (4) generate a proposal to be returned to SNA. In case the call contains multiple alternatives, a Evaluation analysis component is called to perform quantitative analysis and to provide a evaluation rating of the alternatives. The highest valued alternative will be sent back to BU together with SU's constraints as a proposal, which starts a new round of negotiation. The proposal is evaluated in the same way by BNA against its client's constraints. A counter proposal to SNA may be generated. This process of proposal and counter-proposal will continue until an agreement is reached or either side terminates the negotiation process.

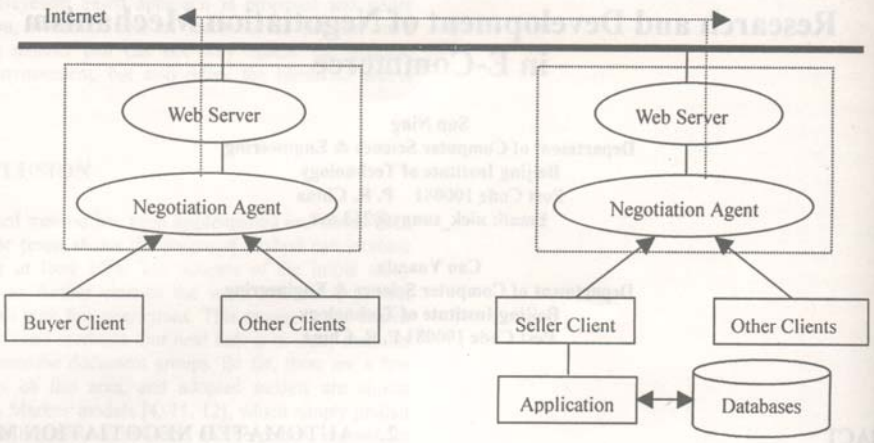


Figure 1: Automated Negotiation Model in E-commerce

3. NEGOTIATION AGENT

The architecture of negotiation agent is shown in figure 2 .It includes five parts: attribute and contraint definition, negotiation strategies, constraint satisfaction check, proposal evaluation, and communication interface.

In order for negotiation agents to negotiation with each other on behalf of their clients, Web-based GUI tools are used for buyers to define requirements and constraints of the goods or services he wants to acquire and for sellers to define the information and constraints related to the goods and services he provides. A client can also define a set of negotiation rules, which specify the negotiation strategies or tactics to be followed when constraints are violated during the negotiation process. Additionally, he can specify the preference scor functions and weights of attributes of goods or service to used for the evaluation analyses of alternative combinatio of negotiation conditions.A constraint satisfaction-processi component is used to check a negotiation proposal counter-proposal against defined requirements constraints. As a constraint violation occurs,strategy rules applied to relax constraints automatically.The relax constraints are used as the basis to form counter-proposals.

A evaluation analysis component performs quantitati evaluation on alternative combination of negotiati proposals to make the optimal selection and generate coun proposal. Negotiation agents communicate with each other sending and receiving messages. Proposal is embedded message as its content for transmission.

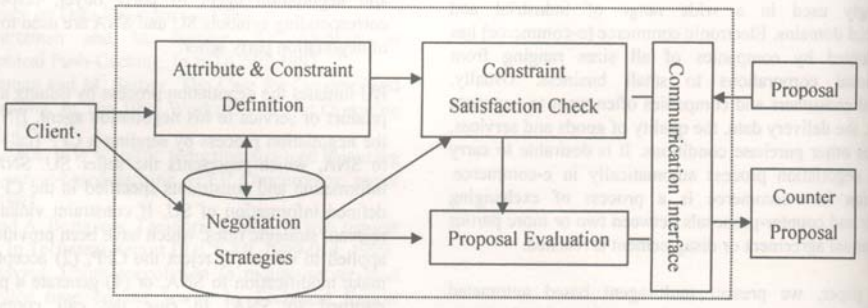


Figure 2: Architecture of Negotiation Agent

4. NEGOTIATION PROCESS AND PROPOSAL EVALUATION MODEL

4.1 NEGOTIATION PROPOSAL

Existing negotiation systems allow only constant values to be specified in a proposal.For example, in a proposal describing a task allocation scenario, the negotiated task must have fixed values for resource requirements, time deadline, cost, etc. The disadvantages of this restriction are obvious. First, a client may not have the domain knowledge to give a value for a certain attribute. Second, a client may not have enough knowledge to compute values that depend on other unknown values.In order to avoid these shortcomings, we define attributes,constraints and inter-attribute contraints in proposals. The following example shows an initial proposal issued by a buyer of purchasing coats:

```
CLASS Proposal {
ATTRIBUTE-CONSTRAINT:
coat_style String   ENUMERATION {chinese_style};
quality String ENUMERATION {wool,cotton};
```

size String ENUMERATION {small, medium, large};
color String ENUMERATION {yellow, green, gray, red, cyan};
unit price Float UNSETTLED;
deliver day Integer RANGE[5…13];
quantity Integer RANGE [300…500];
warranty String UNSETTLED;
INTER-ATTRIBUTE-CONSTRAINT:
Constraint1 color= "cyan" means quantity<=400;
Constraint2 quality= "cotton" means unit price <= 300.00;
}

Each attribute value is of a particular type ,e.g. String, Integer. An attribute constraint is specified by enumerating a set of possible values or by a value range. In addition, attributes whose values depend on other values and cannot be determined until the negotiation is under way are marked as UNSETTLED. Inter-attribute constraints describe the relationships between two or more attributes. Since range and enumeration are used to specify the constraints of some or all of the attributes, a proposal defines many combinations of data values which increases the flexibility and efficiency of proposals and lower the communication cost. A proposal is created and stored as an instance of a proposal object class.

4.2 NEGOTIATION PROCESS

After receiving a proposal, a negotiation agent has the following choices: accept it, respond with a counter-proposal,reject it with an optional explanation, or terminate the negotiation. In order to make a decision, the negotiation agent follows the following general proposal processing procedure as shown in figure 3.

Received proposal is processed by SNA against the defined information. Attribute constraints and inter-attribute constraints are checked sequentially. If constraint violations are found then strategy rules are used to relax the constraint. After the constraint verification, SNA can decide to reject or accept the counter-proposal, or generate another counter-proposal to be sent back to BNA according to the evaluation arithmetic and relaxed constraints. This back-and-forth process continues until a mutual agreement is achieved or either side terminates the negotiation process.

A combination of cooperative and competitive negotiation strategies is used in negotiation process. "Cooperative" means that they are willing to relax their constraints when constraint violations are detected. "Competitive" means that their bottom-line values specified in their negotiation rules will not be made known to their opponents and, if a bottom-line value is reached and the violation still exists, the proposal being checked will be rejected. The other party will be informed of the reason for the reject. It is the other party's turn to decide if a concession should be made. When the counter-proposal arrives at the buyer's BNA, it will go through the same processing procedure except that the defined constraints, strategic rules, and the proposal evaluation arithmetic of the BU are used.

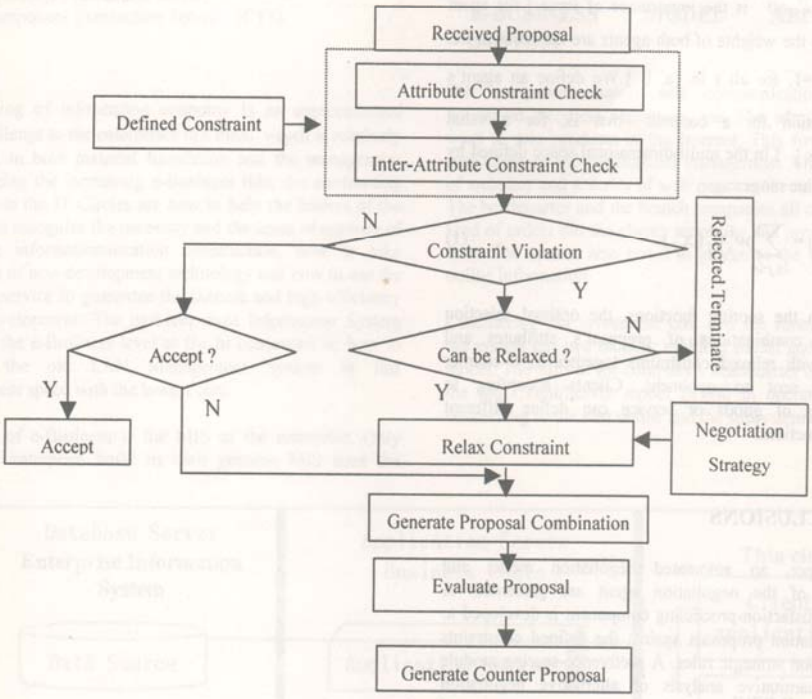


Figure 3: Negotiation Process

4.3. PROPOSAL EVALUATION MODEL

Negotiation is a complex decision making process. The decisions may include not only the rejection or acceptance of a proposal, but also the evaluation of and the selection from multiple choices. The latter is needed in the following situations:

1) A client receives a proposal, which specifies a number of alternative negotiation conditions.
2) A client may conduct negotiations simultaneously with multiple parties.
3) A single proposal may contain a large number of value combinations, which need to be evaluated to make the optimal selection when determining the final agreement or forming a counter-proposal .We have developed a quantitative evaluation arithmetic to evaluate received proposals.

Let i (i∈{a , b}) represent the negotiating agents and j (j ∈ {1,...,n}) the issues under negotiation. Let $x_j^i \in [\min_j^i , \max_j^i]$ be a value for issue j .Here we consider issues for which negotiation amounts to determining a value between a delimited range. Each agent has a scoring function $v_j^i : [\min_j^i , \max_j^i] \rightarrow [0,1]$ that gives the score agent i assigns to a value of issue j in the range of its acceptable values. For convenience, scores are kept in the interval [0 ,1]. The next element of the model is the relevant importance that an agent assigns to each issue under negotiation. $w_j^i > 0$ is the importance of issue j for agent i .We assume the weights of both agents are normalized, i.e. $\sum_{1 \le j \le n} w_j^i = 1$, for all i in {a, b }.We define an agent's scoring function .for a contract –that is, for a value $X=(x_1, \ldots, x_n)$ in the multi-dimensional space defined by the issues' value ranges:

$$V^i(X) = \sum_{1 \le j \le n} w_j^i v_j^i(x_j^i) \qquad (1)$$

According to the scoring functions, the optimal selection from many combinations of proposal's attributes and constraints with relaxed constraints together as a counter proposal is sent to opponent. Clients according to characteristics of goods or service can define different evaluation functions.

5. CONCLUSIONS

In this paper, an automated negotiation model and architecture of the negotiation agent are presented. A constraint satisfaction-processing component is developed to verify negotiation proposals against the defined constraints and negotiation strategic rules. A preference-scoring module performs quantitative analysis of alternative negotiation proposals. Object-oriented knowledge representation and distributed agent technology are applied to the design and implementation of a prototype e-commerce system to demonstrate automated negotiations among buyers and sellers. Experimental result shows the methods are valid and effective. However, some aspects, such as the selection of scoring functions in evaluation arithmetic, need further investigation and improvement.

REFERENCES

[1] H.Raiffa. The Art and Science of Negotiation . Harvard University Press, Cambridge, USA, 1982.
[2] H.Mueller. Negotiation principles.In G.M.P. O'Hare and N.R.Jennings, editors, Foundations of Distributed Artificial Intelligence, Sixth-Generation Computer Technology Series, pages 211 –229, New York, 1996. John Wiley.
[3] J.S.Rosenschein and G.Zlotkin. Rules of Encounter. The MIT Press, Cambridge, USA, 1994.
[4]J.Koistinen and A.Seetharaman, "Worth-Based Multi-Category Quality-of-Service Negotiation in Distributed Object Infrastructures", HP Technical Report, 1998.

The Design and Realization of the TJP2000 Electronic Business System Prototype *

Zhang Jihua, Li Gwangwan, Li Bushang, Chen Qiping, Li Xiaoyu
Department of Computer Engineering, Wuyi University
Jiangmen, Guangdong P.R. China
jhzh@wyu.edu.cn gwli@letterbox.wyu.edu.cn

ABSTRACT

For the companies, facing the increasing electronic business, the ultimate challenge is: how to based on the old intranet Management Information System, using the lowest cost and the highest development speed and the new development technology, build an e-Business application system, upgrade the company manner from intranet or manual process to the e-Business level. This paper is study above such problems and developing an e-Business system prototype.

The TJP2000 e-Business Application System Prototype implementation supports two categories of users in the technology: external users, the customers access the company's website and buy anything using a web browser; internal users, the employees of TJP Company maintain daily routine in the intranet or from remote terminal.

Keywords: E-Business,Prototype, Component Object Model (COM), Transaction Server, Jaguar Component Transaction Server (CTS)

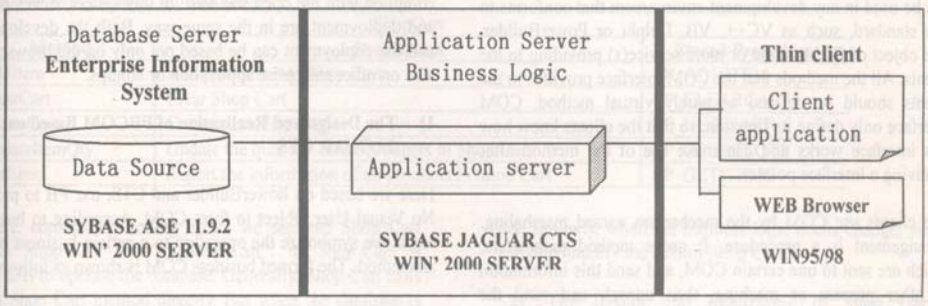
The coming of information economy is an unprecedented sharp challenge to the enterprises in China, which is relatively backward in both material foundation and the management level. Facing the increasing e-Business tide, the confronting problems to the IT Circles are how to help the leaders of the enterprises recognize the necessity and the sense of urgency of enterprise informationalization construction, how to take advantage of new development technology and how to use the excellent service to guarantee the success and high efficiency of the development. The problem most Information System (MIS) to the e-Business level at the hi concerned is: how to upgrade the old LAN Management System in fast development speed with the lowest cost.

The base of e-Business is the MIS of the enterprise. Only when the enterprise build its own genuine MIS does the e-Business has its real meaning. We cannot imagine how an enterprise, which is chaotic in the management and basic data and cannot search its own information, can carry on the real information exchange with other enterprises and earn profits. On this basis, we framed the TJP2000 e-Business System Prototype and completed it in about three months by using component-based development model and multitier distributed application model technology. It incorporates the enterprises internal MIS based on Internet/Intranet 3-tier model application and e-Shop website application. The remote branch companies and the employees on business trip can operate the MIS as they do in headquarter. Other companies and clients can directly inquire and buy the products. Together with the Certificate Authentication system and online payment system, it will put the direct trade in Internet and become a truly e-Business system. Following is a brief introduction to the development process of the TJP2000.

I. THE FUNCTIONAL DESIGN OF TJP2000 E-BUSINESS MODEL APPLICATION SYSTEM

TJP companies mainly sale communication and small household electrical appliances, etc. The administrative rank want to sale products in the Internet. This first requires the realization of informationized management. The construction of websites and a series of web pages is the next thing to do. The headquarter and the branch companies all can know what kind of orders did the clients set online and provide service in time. The system also needs to guarantee the security of the online information.

Considering the investment cost and the running efficiency, the system has two models: multitier model and Client/Server model. The MIS of the company's headquarter designed to use the old Client/Server model is still in operation. The part concerning to the sale online and the subsystem of branch



1) The Management Information System Based on The Intranet

company in other place are designed to use the new multitier model. The following is the sketch map of the multitier model. (The sketch map of C/S model is omitted.)

The intranet MIS of the company which consist subsystems such as basic information, produce management, sale management for local, storehouse management, the employee management, statistics, inquiry and report.

Remote dialup service with encryption function is provided, so that the branch companies and the employees on business trip can share the resources of the system safely from long-distance.

2) The Security of the System

Virtual Private Network (VPN) technology. The remote VPN terminal connect the network by dialup link to the headquarter's VPN Server, then link to the Application Server by the VPN Server and visit the Database by the Application Server. Route technology

3) TJP e-Shop Subsystem and Its Function

- The website provides shopping cart to the shopper when they get in;
- The website provides shopping references to the shopper by giving the list of the latest products and most popular products;
- The website should keep record of the chosen products as a reference for future purchasing;
- The shopper can apply for a membership on the website and get a unique membership card number;
- The shopper can glance over the products and buy their favorites without identification;
- The shopper can revise the quantity of the products in the shopping cart;
- Non-members cannot set any order even after they choose the products. Members will be asked to fill the form in details such as the method of payment and delivery. After they submit the form, they just wait to be informed by the company.

II. THE SYNOPSIS OF THE XCOMPONENT TECHNOLOGY

Component Object Model (COM) is an object-oriented programming specification, which is independent from programming language. COM is a binary system standard. It can be used in any development environment that conforms to this standard, such as VC++, VB, Delphi or PowerBuilder. The object of COM is one or more service(s) providing to the clients. All the methods that the COM interface provides to the clients should be defined as purely virtual method. COM interface only define its function, so that the clients know how this interface works and can make use of the method after receiving a interface pointer.

The clients use COM by the mechanism named marshaling. Arrangement is a procedure. It packs method parameters, which are sent to one certain COM, and send this information to other process or machine, then unpack and send the parameter to the method that transferring the COM, at last pack and unpack the answer to clients.

III. The Principle and Application of Jaguar CTS

1) The Charateristics and the Main Function of Jaguar CTS

Charateristics: enable the common COM to transaction process; coordinate distributed process; supervise the server process pool and thread pool that is equal to cache; support network topology.

Main Functions: supervise the connection of database; manage DCOM; coordinate the transaction process; guarantee the security of access to COM and so on. CTS are a bridge between COM and database. All the COMs in the middle-tier are controlled by CTS. When more than one user visit the COM at the same time, CTS will put the COM into the thread pool and supervise it automatically in order to avoid the network jam .

2) The Use of Jaguar CTS

All the COMs running in the server terminal should register in CTS and finished by Jaguar Manager. The process is as follows:

Establish a package. The package is a collection of all the COMs running in one process. Different COM in different package runs in process-isolated. The COMs connecting to one certain database should be organized into the same package.

Add COMs to the package. Choose package file, which the COMs will be, installed in, click the button "Install New Component(s)" then choose the DLL to be registered. Whenever you re-compile the ActiveX DLL which is already registered, you should reinstall the information in the CTS. Distribute the package. From the package, export the EXE file the information that the clients accessing server COM. The file includes the location of COMs, limits of authority and so on.

IV. THE COM DESIGN AND REALIZATION OF MIDDLE-TIER IN INTRANET MULTITIER MODEL

The application of Intranet represents a new application system model. It is an application model based on n-tier model, component-based development model technology and complied with the open standard. In this system, development and deployment are in the same way. Both the development and the deployment can be based not only on the browser but also on other enterprise application or laptops.

1) The Design and Realization of PBCOM Based on JAGUAR CTS

Here we based on PowerBuilder and CTS, use PB to provide No Visual User Object to form COM. According to business logic, we symbolize the operation to a certain business object as method. The formed business COM is shown as follows.

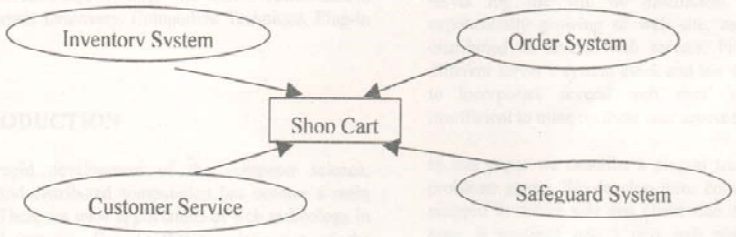
2) Deploy PBCOM to be Run in CTS

To deploy DLL file of PBCOM with the PBCOM painter of PB; Copy DLL to the Windows'2000 server, which is Installed with CTS server, Establish a new empty package in CTS and add PBCOM into it.

Business Object	Description of Operation	Class Method
n_connect	Connect to database	of_connect
n_datawindow	Update a Data Window object	of_update
	Retrieve a Data Window object	of_retrieve
	Retrieve a Data Window object (with parameter)	of_retrieve2
N_sell	Get valid output-notice number	of_get_new_out_no
	Update output table and detail table	of_update
	Update state of invoice	of_update_invo_state
N_store	Update the stock of acertain product	of_update_prod_stocks
	Initialize the stock of acertain product	of_init_prod_stocks
	Check the sufficiency of a certain product	of_check_prod_stocks
N_cont	Update the contract state	of_update_contbook_state
	Update the saled amount of acertain product in certain contract	of_update_cont_prod_num

V. THE DESIGN AND REALIZATION OF E-SHOP WEBSITE

Establish e-Shop and maintain a series of choices made by the shopper. The choices can be checked and changed at anytime. together with the Inventory System, Order System, Customer Service System and the Safeguard System form the "Shop Cart" application program that shown as follows.



1) The Function of The Shop Cart

Product units can be added to the Shop Cart; The Product units in the Shop Cart can be saved in more than one session

Customers can see the information of product units in Shop Cart;

Customers can change the product units in Shop Cart: to delete one or all of the products;

Support boundless product units.

2) The Design of The Shop Cart

We use Java technology to realize business logic; that is to say, we use Enterprise Java Beans (EJB) technology to achieve all the function of Shop Cart. We access database by the store procedure to and use script to combine the Shop Cart to the organization of Java Server Page (JSP) as listed in the following table.

ShopCart		
Method	**Description**	**Stored Procedure**
AddItem	Add products into Shop Cart	SP_ADDITEM
ClearCart	Clear Shop Cart	SP_CLEARCART
CreateCart	Create new Shop Cart	SP_NEWCART
UpdateItemQty	Update the quantity of the products in Shop Cart	SP_UPDATEQTY
GetItems	Report the information of all products in Shop Cart	SP_GETITEMS

Here, considering coding cycle, we separate Shop Cart into "Shop Cart" and "db_Shop Cart". "db_Shop Cart" is an EJB to operate the database. Generally, Shop Cart calls db_Shop Cart method directly. But when the database is changed, changing the EJB of db_Shop Cart directly will do. Similarly, when the business logic needs change, you can only modify the EJB of Shop Cart.

2) The Design and Realization of Shop Cart Environment

Windows 2000 Server with JAGUAR CTS installed;
Java Class file in Shop Cart EJB has registered in CTS;
Allocate Shop Cart object in JSP homepage by Java..

More detailed information about TJP2000 Electronic Business System Prototypecan be obtained from website: http://www.tjp2000.com

REFERENCES

[1] Li Zhijun , Li Fei & Xiao Yongbo, 《Sybase Component Transaction Server》, Publishing House of Electronics Industry, 2000.
[2] Analysis and solution about Simple Chinese problem in java coding technology, Duan Minghuihttp://d23xapp2.cn.ibm.com/developerWorks/java/java_chinese/index.shtml
[3] Creating a Simple PowerJ EJB Application, Martyn Mallick, and Technical Evangelist, Sybase, Inc. http://my.sybase.com/detail?id=100238
[4] Creating and Deploying Java Servlets with PowerJ 3.0 and EAServer 3.0, sybase inchttp://my.sybase.com/detail?id=1001350
[5] Using Stored Procedures with PowerJ, sybase inc http://my.sybase.com/detail/1,3693,1010822,00.html

Obtaining the User Access Information from Client Side

Wu Xinling Wang Zebing Feng Yan
Computer Science and Engineering Department, Zhejiang University
Hangzhou, Zhejiang, China

Email: wuxinling@263.net ; wzb@mail.hz.zj.cn ; fyan2001@263.net

ABSTRACT

With the rapid development of Internet, web-based Applications experience a great boom and web sites offering personalized service become the focus of the current research. Web usage mining has made many advances in this field. However, the tradition pattern of obtaining the user access information from the web server is unable to meet the demands of supplying accurate and adequate user formation. The situation get worse as the Internet grows larger and many web sites distribute at several different web servers. The usage mining of such web sites is difficult. This paper presents a new method for such situations by using a plug-in method that can obtain user access information from client side directly and send it to a midway server, namely usage mining server. The information thus obtained is accurate, adequate and comprehensive. As all the modules of this application are implemented in the form of components, it is easy to reuse the application in distributed systems.

Keywords: Web Usage Mining, Transaction Identification, Frequent Itemset Discovery, Component Technique, Plug-in Method

1. INTRODUCTION

With the rapid development of the computer science, Web-based and distributed computation has become a main trend of IT. There are wide applications of web technology in Internet and intranet. Due to the popularization of the distributed computation, it's very valuable to develop the web-based applications used in distributed environment. On the other hand, the study of data mining based on web became a focus of researcher. Web usage mining [1,7], automatic discovery of user access patterns, can automatically adjust the site's presentation for an individual user. For example, the site recommends the new pages to users based on their access behavior, which is wide applied in the field of e-commerce [3], adaptive web site [6] and others. A generalized architecture of web usage mining is depicted in Figure 1.

There are many techniques developed to discover the user access pattern, such as: Frequent Itemset discovery [8], mining frequent traversal trees (MFTT)[5], clustering [3]. The first step of implementing these methods is Data Preparation, which is to capture the user access behavior. The traditional method uses the web server's log file [2], which restricted to the web server side, so there may exist much information that is important but neglected by web server. For instance, firstly, when a client request the web page through one proxy agent, the web server only can have the information of the proxy agent instead of this client. Therefore, the single user access pattern cannot be obtained. Secondly, most of web browsers cache the pages that have been requested. As a result, when a user hits the "back" button, the cached page is displayed and the web server is not aware of the repeated page access and the record of web server log file will be insufficient. Thirdly, with the exponentially growing of web site, one web site can be distributed in several web servers. For the difference of different server's system clock and log style etc, it's difficult to incorporate several web sites' log files. It's also insufficient to mine on these user access behaviors.

In this paper we describe a plug-in technique to solve the problems above. We develop three components that can be plugged in server side and client side directly. First, a web page is wrapped into a new web page by a server-side wrapping module. Then, the unwrapping component in the client side is called to unwrap the new page into the original web page. It also informs the usage-mining server (UMS), the other component of our design, of the client browser behavior. By this method, the web page will not be displayed to user unless the unwrapping module is called and the user behavior is reported to UMS. We emphasize the method in following section. In section 2, the system architecture is

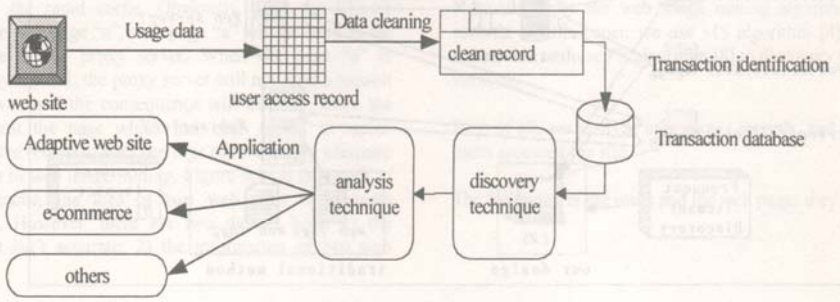


Figure 1: Web Usage Mining Architecture

depicted and every component of the system is also described. In section 3, we display an experiment to test our plug-in method. The traditional method and our design are used in the same scenario. At the end we have a conclusion that our design is available and superior.

2. DESIGN AND IMPLEMENTATION OF SYSTEM

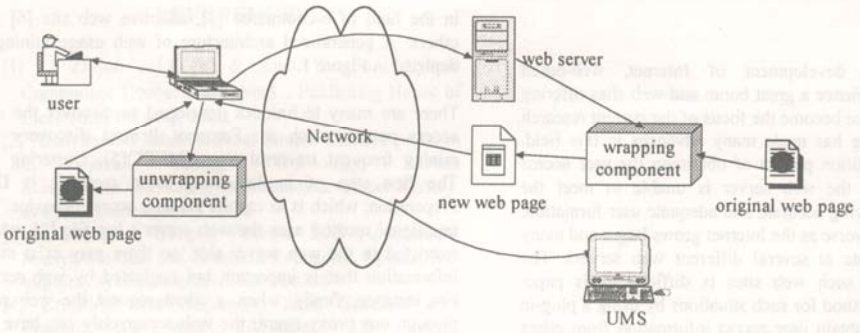


Figure 2: The Process of System

Now, we analyze the process of the system implementation that is depicted in Figure 2. At the first step, a client browser is used to request a web page from the web server. Then, the new web page, wrapped by the wrapping component is sent to the client browser, which detects the unwrapping component should be called to convert the new page to the original page and to inform UMS of the behavior of this client browser. Finally, the content of this page is displayed to user by the browser.

Obviously, there are three components we should develop: the Wrapping Component, the Unwrapping Component and the User Pattern Mining Component .The wrapping component and the UMS component are used in server side and the unwrapping component should be run in client side.

The Wrapping component converts original document to a new document. It has two tasks. First, it should encipher the original document to avoid the client bypass the collection of user access behavior. So the new document should include two parts: one is the code of invoking unwrapping component processing; the other is the actual text of enciphered document.

The unwrapping component should be downloaded to the client side and run there.

The User Pattern Mining Component mainly collects the information of user access behavior, and loads them into database. Our user access record includes 4 parts: user IP address, user host name, access begin time, the URL of web page. Evidently, the information of user access behavior is accurate and adequate compared to logs in an ordinary web server.

3. APPLICATION

The web system depicted in figure 3 use the Microsoft IE browser and the IIS web server. We obtain the user access information by the plug-in method, and implement the mining method on it. The result is compared with the traditional method.

Consider the scenario in Figure 3, where three clients access the web site MYWEB, which has two web servers: web server1 and web server2w The Client1 and Client2 access MYWEB through the proxy server and the Client3 accesses MYWEB directly. Every web server has its log file. Traditional method applies directly on these log files. Our design adds a server to collect the user access information, and implement the mining algorithm on it.

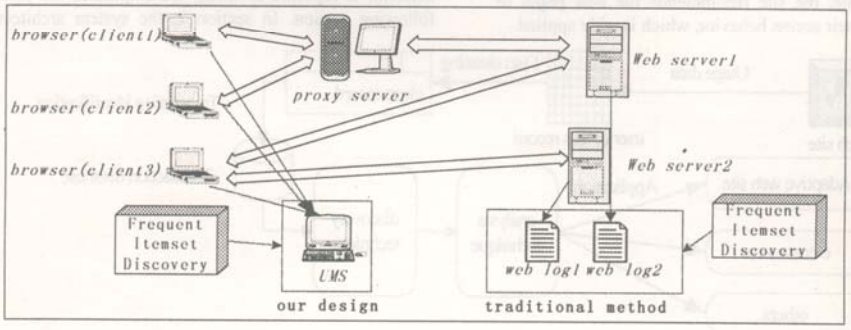
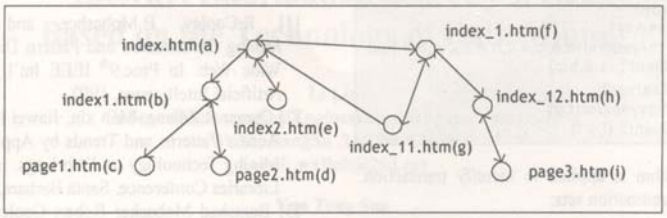


Figure 3: Scenario for our experimentation

In Figure 4, the part of MYWEB architecture is depicted.



Note: The character in bracket is the abbreviative label of web pages

Figure 4: The Part of Web Site Architecture

Web server 1 logs

id	User host name	url
1	Proxy server	a
2	Proxy server	b
3	Proxy server	c
5	Client3	a
6	Proxy server	e
7	Client3	f
8	Client3	h
9	Client3	f
10	Client3	a
11	Client3	b
12	Client3	c

Web server 2 logs

id	User host name	url
1	Proxy server	f
2	Proxy server	g
3	Client3	f
4	Client3	g

Figure5 (a): when the proxy server and rapid cache exist

Web server 1 logs

id	User host name	url
1	Proxy server	a
2	Proxy server	a
3	Proxy server	b
4	Proxy server	a
5	Client3	a
6	Proxy server	c
7	Proxy server	b
8	Proxy server	a
9	Proxy server	e
10	Proxy server	b
11	Proxy server	c
12	Proxy server	b
13	Proxy server	a
14	Client3	a
15	Proxy server	b
16	Client3	b
17	Proxy server	c
18	Client3	c

Web server 2 logs

id	User host name	url
1	Proxy server	f
2	Proxy server	g
3	Client3	f
4	Client3	g
5	Client3	f

Figure5 (b) when the rapid cache doesn't exist

UMS records

id	User host name	url
1	Client1	a
2	Client1	f
3	Client1	g
4	Client1	a
5	Client1	b
6	Client2	a
7	Client3	a
8	Client1	c
9	Client1	b
10	Client1	a
11	Client1	e
12	Client2	b
13	Client2	c
14	Client2	b
15	Client2	a
16	Client3	f
17	Client3	g
18	Client3	f
19	Client3	a
20	Client2	b
21	Client3	b
22	Client2	c
23	Client3	c

Figure5 (c) the records of UMS when the plug-in method is applied

Figure 5: User Access Records

Three client access record sets are depicted in the Figure 5.

Figure 5(a) describes the records when there exist a proxy server and the rapid cache. Obviously, after the Client1 accesses the web page "a", web page "a" will be store in the rapid cache in the proxy server. When the page "a" is requested by Client2, the proxy server will not send a request to web server, and the consequence will be same when the client request the page which has been stored in cache. Therefore, the record of web site logs cannot supply adequate information to web usage mining. Figure 5(b) is the result of no rapid cache, the logs of two web sites is adequate themselves. However, there are two defects here: 1) the information isn't accurate; 2) the information of two web sites cannot be incorporated. Figure 5(c) displays the information of our usage mining server. (The user host name is omitted for the convenience of analysis.)

Now we apply the web usage mining algorithm on these records. In this paper, we use MS algorithm [4] to identify transaction and use FS algorithm [8] to discovery the frequent itemsets.

First of all, we analyze user access records, and identify the users accessed the site.

The following is the users and the web pages they accessed:

```
UMS:
        Client1:{a,f,g,a,b,c,b,a,e}
        Client2:{a,b,c,b,a,b,c}
        Client3:{a,f,g,f,a,b,c}
Traditional method:
        Web server1:
            Proxyserver{a,a,b,a,c,b,a,e,b,c,b,a,b,c}
            Client3:{a,a,b,c}
        Web server2:
            Proxyserver{f,g}
            Client3:{f,g,f}
```

Secondly, MS algorithm is applied to identify transaction. The following is the transaction sets:

```
UMS:
{{a,f,g},{a,b,c},{b,a,e},{a,b,c},{b,a},{b,c},{a,f,g},{a,b,c},{a,f,g},{f,a,b,c}}
Traditional method:
Web Server1:{{a},{a,b},{a,c,b},{a,e,b,c},{b,a},{b,c},{a},{a,b,c}}
Web Server2:{{f,g},{f,g},{f}}
```

Finally, the minimal support 0.25 is used, and the result after FS algorithm applied:

```
The maximal frequent itemsets of UMS is:
    {a,f,g} and {a,b,c}
The maximal frequent itemsets of traditional method is :
    {a,b},{b,c} and {f,g}.
```

Obviously, the frequent item sets obtained by traditional method aren't adequate. We can make a conclusion from the set {a, f, g} and {a, b, c} that when a user access page "a", he usually accesses "g" through "f" or access "b" through "c". Then there may be a guess: we should maybe add some new hyperlinks from "a" to "g" and from "a" to "c". A simple conclusion is obtained by the usage mining. However, we cannot catch this conclusion by the traditional method. There are two reasons: first, there exists a proxy server. In a Web server log, all requests from a proxy server have the same identifier, even though the requests potentially represent more than one user. For instance, in figure 5(c), the 5th and 6th records that weren't produced by one user would look like one user in figure 5(b), so the traversal path of Client1 {a, b, c} is cut off. Finally the frequent item set {a, b, c} cannot be found out. Secondly, the record of two web servers cannot be incorporated. Therefore, the traversal path {a, f} cannot be connect to traversal path {f, g}. Our design can solve this problem.

4. CONCLUSION

This paper presents a new method that is able to accurately connect the user access information using plug-in method. This new method solves three problems effectively: the proxy server, the rapid cache and multiple web servers. Another major advantage of plug-in is its component-based implementation, which indicates its easy adaptation to the distributed systems. There are many web sites within the intranets of universities and corporations. The method we introduced can be used to incorporate these web sites' access information. What's more, the separation of usage mining server from web server can alleviate the web server's work.
In summary, the method we designed can supply accurate and adequate user access information, and improve the accuracy of web usage mining, and accelerate the pace of web mining application.

REFERENCE

[1] R.Cooley, B.Mobasher, and J.Srivastava. Web Mining :Information and Pattern Discovery on the World Wide Web. In Proc.9th IEEE Int'l Conf. On Tools with Artificial Intelligence, 1997.

[2] Osmar R.Zaiane, Man xin, Jiawei Han. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. In: Advances in Difital Libraries Conference. Santa Barbara, CA, 1998, pp.19-29

[3] Bamshad Mobasher Robert Cooley, Jaideep Srivastava. Creating Adaptive Web Sites Through Usage-Based Clustering of URLs. In Knowledge and Data Engineering Workshop, 1999

[4] R.Cooley, B.Mobasher, and J.Srivastava. Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns. In Knowledge and Data Engineering Workshop, pages 2-9,Newport Beach, CA. 1997. IEEE

[5] Xuemin Lin& Xiaomei Zhou Yuh-Chi Lin. Efficiently Mining Tree Traversal Patterns in a Web Environment. IEEE, 1998, 115-117.

[6] Mike Perkowitz, Oren Etzioni. Adaptive Web Sites: an AI Challenge, in: Proc. IJCAI-97, Nagoya, Japan, 1997.

[7] Chen, M.S., Park, J.S. and Yu, P.S. Data mining for path traversal patterns in a Web environment. In Proceedings of 16th International Conference on Distributed Computing Systems, 1996.

[8] Ming-Syan Chen, Jong Soo Park and Philip S.Yu. Data Mining for Path Traversal Patterns in a Web Environment Proceedings of the 16th ICDCS 1996 IEEE 385-392.

Internet Information Search Service Based on the Technology of Data Mining

Li Liu
College of Information Engineering, Jiangnan University
WuXi, JiangSu ,214000, China
Email: wxliuli@263.net

Yan Tong Sun
College of Information Engineering, Jiangnan University
WuXi, JiangSu ,214000, China
Email: ytsun123@163.com

ABSTRACT

In this paper, we compare and analyze existing Internet information search service pattern, and describe a distributed metasearch engine model. After having analyzed existing data mining technology, we choose decision tree technology to establish and to optimize this model.

Keywords: Data Mining, Decision Tree, Distribution, Metasearch Engine, and Information Search Service

1. INTRODUCTION

The Internet has rapidly developed recent years. The information resources on the Internet have proliferated. Highly praising the richness and variety of the Internet, users frequently feel worried about not being able to locate needed information promptly. It is against the background of enormous demand that Internet information search technology comes into being. We name these application programs that offer Internet information search service as search engine.

If we integrate information search powers of existing search engines on the Internet, we can, according to the type of user's inquiry request and real-time condition of network, send user's inquiries to relevant search engine to process. In this way, we can focus on improving distributed management strategy of search engines, filtering and integrating the search results from these search engines, in order to improve the quality of information search service. The distributed metasearch engine model described in this paper is based on this thought.

Each search engine resembles a node in the distributed structure, whose work state and information service quality change continuously. Therefore, the model's distribution strategy should also be dynamic. If we measure each search engine against its historical data of information service and user's feedback, we can optimize this search engine model. The data-mining module that adopts decision tree technology does the work of digging distribution strategy from feedback, establishing and optimizing this model.

2. DISTRIBUTED METASEARCH ENGINE

Comparison of Existing Search Engine Technologies

Presently, search engine technology is developing comparatively fast. There are lots of Internet search engine such as Yahoo!, Excite, Google, Net of Beijing University, etc. The technologies in this field can be classified into the following groups[1]:

(1) Search Engine Based on Catalog:
Information collected by search engine is divided into several classes. The biggest problem of this kind of search engine is that if the information you are seeking for has not corresponding category, the engine can't carry out the search. The typical search engines based on catalog are Yahoo and Magellan.

(2) Search Engine Based On Robot:
It starts searching from a group of known documents, ascertaining the new retrieval point through hyperlink. They have been criticized frequently as unsafe and causing heavy network and server loads. The typical search engine based on robot is AltaVista.

(3) Search Engine Based on Customer:
This kind of search engine is Web customer end software. It searches from a group of known documents, retrieves the documents on WWW and then delivers them. The searching is real-time and it will get latest information. However, its search speed is lower, and network or server loads are too heavy.

(4) Metasearch Engine:
Metasearch engine sends user's inquiries to other search engines. It focuses on improving user interface and filtering the relevant document taken from other search engines. Metasearch engine are simply constructed, just as Metacrawler.

Model Frame of Distributed Metasearch Engine

Existing metasearch engine just accepts user's search request and concurrently sends them to other famous search engines. Then it collects returning results and delivers them to the user. Now, available metasearch engine merely offers users the unified interface to several search engines, but does not solve the problem at hand.

After our comparison and analysis of the above kinds of search engines, we have discovered the fact that each kind of search engine has its particular advantages and priority, but certain limitation on information search scope. For instance, in terms of search scope, Yippee searches Chinese network address all over the world, mainly in Chinese mainland. Tonghua search engine collects the Chinese network address, covering the mainland, Hong Kong, Macao and Singapore. The search engine of the Net of Beijing University chiefly includes CERNET information. In terms of search range of content, the Net of Beijing University search engine prioritizes particularly on research and education, whereas, China Economy Net search engine centers on commerce and industry.

Based on the above comparison and analysis, we first classify search engines with its priorities and advantage. The classification should be a broad one, with such divisions as educational research, business and economy, cultural entertainment, health service, shopping, etc. For instance, the Net of Beijing University is attributed to the category of education and research; China Economy Net belongs to that of finance and economy. Comprehensive search engine may be fall into more than one category. Secondly, search engines in the same category are set with different prior ranks, which are decided by the result information of users' feedback. In the distributed metasearch engine model, each search engine corresponds to a node of the distributed system. Their tasks are users' search requests. The tasks distributed to which search engine depends on the historical record of each search engine. The following chart presents the working process of a distributed metasearch engine:

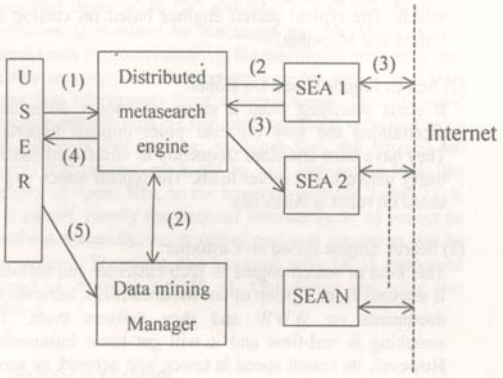


Fig.1 model frame of the distributed metasearch engine

(1) Using WWW browser, the user can visit systematic interface. Selecting an optional category and inputting search keywords, the user will send inquiry request to system.

(2) After receiving user's request, the distributed metasearch engine manager will activate partial search engine agents (Abbr. SEA) to work, under control of the data mining manager.

(3) SEA will translate the standard inquiry request sent by distributed search engine manager into local inquiry request, activate local search engine to seek relevant information. Then, SEA analyses returned HTML page and sends effective information to the distributed metasearch engine manager.

(4) The distributed metasearch engine manager refines and integrates information returned from each SEA, and returns a search result page to the user.

(5) When the user receives result page, he (or she) may express the satisfactory opinion for the search information entry through a feedback button on the page. These information are eventually handled by a data mining manager to judge the working quality of each search engine, and to decide on the distributed search strategy of next request .The more feedback that user offers, the more distributed strategy that system implements meets the user's demand.

Search Engine Agent (SEA)

As a browser, Search engine agent communicates with search engine, which is part of the system to get information resource on the Internet. Each search engine agent is an independent process .It works equally with others under the control of the search engine manager, and sends inquiry results to the manager by way of process communication. So long as we send applications according to HTTP protocol to a Web server, the server will give corresponding reply. The search engine agent will complete two major tasks:

(1) Information inquiry:
SEA connects with a corresponding search engine and translates standard requests into engine's local requests. It takes the role of shielding engine discrepancy, and offers place transparency. So, studying inquiry requests parameters of each search engine and its result page are the basic work of the system.

(2) HTML document analysis:
Information sent by a Web search engine is described with HTML (Hypertext Markup Language), which can usually be divided into two parts. One is the control identifier, starting from " < " to " > ", including the string between them, such as "< TITLE >" .The other is the string, which can be seen when you browse this HTML page by browser, such as IE, or Netscape. Therefore, we can analyze HTML documents according to the following process: Scanning documents, filtering control identifier, at the same time, adding weight right according to specific identification, such as bold (<B>), and forming an information entry – metadata. The data mining manager will determine the specific form of metadata.

Information Filtration and Integration of Distributed Metasearch Engine Manager

There is large quantity of information sent from search engines. It is common to find repetitive information.

(1) Same URL address:
We will choose the newest information record among them.

(2) Different URL, the same content:
Under most conditions, two different pages will not produce identical abstract information. If abstract information is the same, we consider them as the same

copy. Mirror website and information reprint will cause this kind of condition. In that case, we merge that information; mark a copy of content with several URL addresses that can be visited.

(3) URL with same initial substring:
When using a browser, we often find that lots of result information is under the same directory of certain website. They have the same initial substring of URL address. If that is the case, we can incorporate this kind of information, show with the form of initial substring of the URL address, and refine the return information. Besides, some information is hidden for lower matching degree, unless the user asks to look over it.

According to this kind of condition, we can integrate this kind of information, show with the form of initial substring of the URL address, and refine the return information. Some information is hidden for lower matching degree, unless the user asks to look it over.

3. THE APPLICATION OF DATA MINING TECHNOLOGY

The establishment and management of distributed metasearch engine directly concern the quality that information search service offers. The Internet itself is a big distributed information processing system, in which each service node is a metasearch engine. Since each metasearch engine is managed by a particular company or website, its working state and major service direction are always in changing, which make it difficult to maintain by hand. It is unscientific to predetermine either its prior rank or classification of metasearch engine selected. The quality of information search service also cannot be guaranteed. Digging valuable information in mass data is the advantage of data mining.

Comparison of Existing Data Mining Technology

Data mining has got a fast development in last few years. Technology of nerve network, association rules, clustering and decision tree become major methods in data mining technology [2].

(1) Nerve network:
Data mining based on the nerve network, according to marked behavior, fits especially for distinguishing certain pattern or forecasting certain developing tendency.

(2) Association Rules:
If the value between two or many variables has certain kind of regularity, it is called association. Association rules technology is used to study data set, endeavor to reveal the association among data internal attributes, and to find out the association network that is hidden in database.

(3) Clustering:
The record in database can be divided into a series of meaningful subclasses — clusters, which strengthen people's knowledge of objective reality and is the prerequisite of concept description and drift analysis. The technology of clustering splits one object set into several categories. Objects within each category are similar, but are not similar with those of any other ones.

(4) Decision tree:
Decision tree is divided into classification tree and regression tree. The former deals with dispersed variables, and the latter copes with continuous variables. Usually, a data mining tool permits the user to select split conditions and cut rules, as well as control parameters (the size of minimum node, the depth of biggest tree, and so on) to control the over fitting of decision tree.

Shown in the following table is the comparison of these four kinds of data mining technology in terms of understandability, trainability, enforceability, versatility, usefulness and acquisition:

	Nerve network	Association Rules	Clustering	Decision tree
Understandability	C-	A	B+	A+
Trainability	B-	A	B+	B+
Enforceability	A-	A+	A-	A+
Versatility	A	D	A-	A
Usefulness	A	B	B-	A
Acquisition	A	B	B+	B+

(A is excellent, B is good, C is general, D is bad)

Establishment of Distributed Metasearch Engine by Decision Tree Technology

Different data mining goal or data type needs different technology. Data mining is divided into prescriptive and descriptive according to its goal. The purpose of this topic is to describe a model. Data to be handled in this model is dispersed text type data. So we select the method of making decision tree technology to build and to optimize the model of distributed metasearch engine.

(1) Establishment of training set:
To establish the decision tree of distributed metasearch engine model, training set must be built firstly. The establishment of a decision tree directly depends on the data set, which is called as training set. Therefore, we summarize the results of each engine by hand, fill them into the search engine information service table, and classify search engines initially .We permit comprehensive search engine to be classified into more than one categories.

(2) Recursive generation of decision tree[3]:
A decision tree can be generated by a recursive cut. The data structure of a decision tree is a tree structure, whose root node is an entire data set, and whose sub-node is a test to a unitary variable. Each test divides data set into two or more pieces. The process of bringing up a complete decision tree is an unceasing process to choose classifying field.

In following chart, you can see the basic form of a decision tree for distributed metasearch engine model.
First, information search service is divided into Chinese information search and English information search. Second, each block is classified according to user information request. Third, when building a decision tree, we determine the orders of search engines in a same class by their initial prior ranks. When in maintenance and optimization, the order of them is determined by user's satisfactory degree. Left node has higher prior rank than the right node in the same layer.

Optimization of Distributed Metasearch Engine Model with Decision Tree Technology

In the result pages sent to the user, we have added a user feedback function that collects the user's opinions to the information service. The user expresses his (or her) satisfying or dissatisfying simply through clicking the button. This information classified by classification, search engine, or systematic time, is appended to the user feedback information table in database. The statistical information on this table shows the working states of each search engine on different classification direction. It is simultaneously chosen as test set

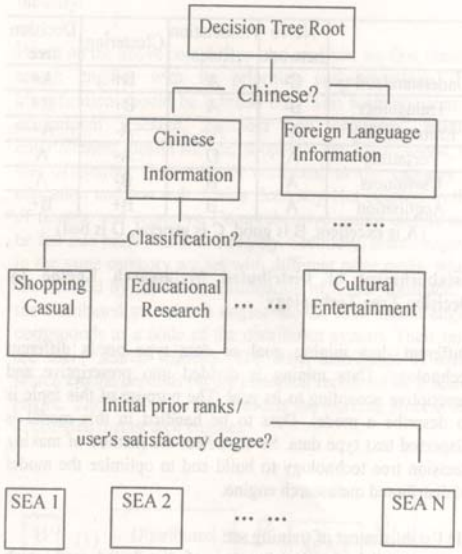


Fig.2 decision tree model of the distributed metasearch engine

data of decision tree.

(1) Cutting branch:

It is not necessary to let a decision tree overgrow. Overgrowth will reduce intelligibility and usability of the decision tree; at the same time it will increase the dependence on historical data. That is to say, the decision tree may show its accuracy to these historical data. However, when it is applied to new data, the accuracy of the decision tree will drops rapidly. We call this situation excessive training. To make decision tree contain universal rules, it is important to prevent from excessive training. Therefore, we need a method to control the growth of decision tree in proper time. We choose the method of setting valve of user's satisfactory opinion rate to cut branch.

(2) The adjustment of branch:

The work of a search engine in certain classification will be cut, if its corresponding user satisfactory rate is lower. Furthermore, the rank of search engine in same classification will also be adjusted according to user's satisfactory rate.

Thus, we can input a user search request in the root node of the decision tree, and get a distributed search strategy fro root to leaf.

4. SUMMARY AND PROSPECT

The information service on the Internet is a potential researc field, in which there are a lot of problems to be wort studying. This paper has put forward a distributed metasearc engine model based on data mining technology and discusse the design thought of each model parts.

In recent senior technological investigation[4], data minin and artificial intelligence are considered as top of five ke technologies, which will exert a far-reaching impact o industry within next three to five years. Parallel processing system and data mining technology rank top two of the te new technologies in the future five years. With the rapid development of technology, such as data acquisition, transitions and saving, large scale systematic user will need more to adopt new technology to dig values besides market, and create new commercial opportunities by a more vast parallel processing system.

REFERENCES

[1] Levy M R. Web Programming in guide. Software, 1998, 28(15): 1581～1604

[2] Jiawei Han, et al. Discovery of multiple-Level association rules from large databases. In: Proc. of VLDB, Zurich, Switzerland, 1995:420-431

[3] Rakesh Agrawal, Tomasz Imielinski, Arun N. Swami: Mining Association Rules between Sets of Items in Large Databases. SIGMOD Conference 1996: 207-216

[4] Kdnuggets Newshttp://www.kdnuggets.com/news/

A Comparison of Service Discovery Protocols

Liang Shuang
College of Computer Science, Huazhong University of Science & Technology
Wuhan, Hubei, 430074, P.R.China
Email: cool1203@263.net

Liang Youming
Fundamental Department of Air Force Radar Academy
Wuhan, Hubei, 430010, P.R.China
Email: liangyouming@21cn.com

Chen Ke
College of Computer Science, Huazhong University of Science & Technology
Wuhan, Hubei, 430074, P.R.China
Email: ckhust@263.net

ABSTRACT

The exploding number of Internet services has made searching and requesting services unmanageable by human beings. And automatic service discovery will urgently be an imperative feature in future network computing environment. "Service discovery" refers to a spontaneous process, in which entities including services and devices automatically "discover" the other entities on the network. This paper discussed the emerging technology of *service discovery*, with emphasis on SLP, Jini, SDS and INS protocols. Common design issues for such protocols are analyzed, prominent features are described, and a careful comparison and their distinguished features are also discussed.

Keywords: Service Discovery, Service Discovery Protocols, Mobile Computing, Jini, SLP, SDS, INS.

1. Introduction

Services in the network and distributed systems are growing at an enormous speed. The mobility and large quantity of the services make it imperative to introduce an auto service discovery and location mechanism into the network infrastructure. For example, a traveling user needs instant connectivity to devices and services at the home office, as well as connectivity to branch-office devices while on the road. Traveling users need to be able to connect to branch office networks without relying on assistance from a branch office's network administrator. With service discovery, the traveling user can automatically search the branch office's local network and access printers, scanners, and other network devices as needed.

The field of service discovery research is relatively a new one, and has received academic attention only very recently. Industrial companies, international organizations and universities have collaborated to have several protocols designed, such as SLP, JINI, INS, UPnP, Salutation and SDS. These protocols are still young, and lots of work is still underway to maturate them. In the following section we will discuss the common issues that are worth considering, section 3 will introduce protocols such as SLP, JINI, INS and SDS, section 4 will compare these protocols in several aspects, and a conclusion is presented in the final section.

2. COMMON ISSUES

Searing and requesting relevant services are the key design issues of service discovery protocols. Although different protocols may have their unique designs and emphasis, problems on discovery model, system architecture service presentation and query language and system security are all addressed.

Basic Discovery Model

Clients must find relevant services, including sufficient information to establish contact and obtain service. There are three basic mechanisms by which this is accomplished: advertisement (push), service request (pull) and a hybrid approach. In a push model, services "advertise" their availability, address, and other necessary information. Clients who receive advertisements may then contact services as they wish. In a pull model, clients may "request" service of some kind, and receive information about services in response. Services or other agents listen for requests and respond appropriately.

System Architecture

Generally, there are two system architectures: server-based and server-less systems. Server-based systems have one or more servers, which accept registrations from services. Clients in these systems query the server to find services. On the other hand, server-less systems are focused on functioning in a smaller environment where a server may not be present. In these systems, clients simply broadcast their requests, and any service matching a given request responds with more information on how the client can interact with it.

Service presentation and query language

A simple, expressible language is needed to be able to ask for services. An expressive language will make service providers able to describe the services that they have. Likewise, clients need to make more powerful queries by taking advantage of

the semantic-rich service descriptions.

Security

Discovery protocols face a real challenge from security. Security includes privacy and authentication. On the one hand, security is definitely needed. For example, my digital camera must not be able to use a printer in the neighbor' s house without permission, and unauthorized parties on the way to my printer must not intercept the pictures from my camera. On the other hand, the desire to be automatic, lightweight, and to minimize computing resource and network usage forces protocols to use very simple schemes.

3. PROTOCOLS

SLP

Service Location Protocol (SLP) comes from Sun Microsystems, and is an IETF standard for "spontaneous" discovery of services. SLP aims to be a vendor-independent standard. It is designed for TCP/IP networks and is scalable up to large enterprise networks.

The SLP architecture consists of three main components:

1. User Agents (UA) perform service discovery, on behalf of the client (user or application);
2. Service Agents (SA) advertise the location and characteristics of services, on behalf of services;
3. Directory Agents (DA) collect service addresses and information received from SAs in their database and respond to service requests from UAs.

The SLP can be implemented in several configurations. In "passive" configuration, the DAs periodically multicast service advertisements. UAs and SAs can also locate DAs using DHCP in "static" configuration; and send query messages to SLP multicast group address in "active" configuration.

SLP can also be implemented without any DA at all, enabling SLP to work with no administration for small networks; while in large enterprise network, DA is used to cache service information to permit service discovery in case of putting too heavy traffic on the network. In the absence of a DA, the UAs and SAs implement all the functions of the DA with multicasts. When one or more DA is present, the protocol is more efficient, as the UA or SA uses unicast messages to the DA.

The SLP defines a "Service URL", which encodes the address, type, and attributes of the service. For example, service:printer:lpr://hostname might be the service URL for a line printer service. Service requests may match according to service type or by attributes. A template and an LDAPv3 predicate specify attribute matching of SLP.

JINI

JINI grew from early work in Java to make distributed computing easier. It intends to make ``network devices'' and ``network computing'' into standard components of everyone's computing environment. Its architecture resembles the SLP, and it is designed as an extension of the Java language.

A JINI network requires at least one copy of the "JINI Lookup Service" (LS), which maintains a service database of the network. The process of service "register and join" and client request are all managed by serial java object communication.

In order to register a service, the service provider must first find the LS by UDP multicast requests. When the LS gets a request, it sends an object back to the server. This object, known as a registrar, acts as a proxy to the lookup service, and runs in the service's JVM (Java Virtual Machine). Any requests that the service provider needs to make of the lookup service are made through this proxy registrar. And the registrar will take a copy of the service object, and storing it on the LS.

The client will go through the same mechanism to request services from the lookup service. But this time it is to request the service object to be copied across to client itself. And there will be a copy of the service object running in the client's JVM. The client can make requests of the service object running in its own JVM. This is clearly a powerful feature, made possible by the single language environment and the mobility of Java code.

INS

Intentional Name Service (INS) is composed of INS services, INS clients and Intentional Name Resolvers (INRs). Each service attaches to an INR and advertises an attribute-value-based service description. Each client communicates with an INR and requests a service using a query expression. INRs self-configure into a spanning-tree overlay network to exchange service descriptions and construct a local cache based on service advertisements.

When a client request arrives at an INR, it is resolved on the basis of the destination name. If the client application has chosen early binding, the INR returns a list of IP addresses corresponding to the name. This is similar to the interface provided by the Internet Domain Name System (DNS), and is useful when services are relatively static. INS applications use the two late binding options— intentional anycast and intentional multicast—to handle more dynamic situations. Here, the binding between the intentional name and network location is made at message delivery time rather than at request resolution time. Thus, INS uses an intentional name not only to locate services, but also to route messages to the appropriate endpoints. This may speed performance when service availability is changing rapidly and make programming much more easily.

INS achieves expressiveness by using expressions called name-specifiers based on a hierarchy of attributes and values which are free-form strings defined by applications.

The name discovery protocol treats name information as soft-state, associated with a lifetime. Such state is kept alive or refreshed whenever newer information becomes available and is discarded when no refresh announcement is received within a lifetime. This choice allows a design where applications may join and leave the system without explicit registration and de-registration, because new names are automatically disseminated and expired names automatically eliminated after a timeout. The approach can make the system extremely robust with minimal manual intervention.

SDS

The SDS is part of the University of California, Berkeley Ninja research project. The SDS is similar to the above discovery protocols, with a number of specific improvements in reliability, scalability, and security.

The SDS consists of five components: SDS servers, services, capability managers, certificate authorities, and clients. SDS servers solicit information from the services and then use it to fulfill client queries.

As a scalability mechanism, SDS servers organize into hierarchical structure; service announcements and client queries are assigned to go to a particular SDS server. If a particular SDS server is overloaded, a new SDS server will be started as a "child" and assigned a portion of the network extent.

SDS also uses a soft-state approach to error recovery and self-healing. Each server is responsible for periodically sending authenticated messages containing a list of the domains that it is responsible for on the well-known global SDS multicast channel. A service listens to the information to determine the correct SDS server, and then multicasts its service descriptions to the proper channel with the proper frequency. The SDS server (and clients) may cache service information. The caches are updated by the periodic announcements or purged based on the lack of them. In this manner, component failures are tolerated in the normal mode of operation rather than addressed through a separate recovery procedure.

The SDS uses XML to describe both service descriptions and client queries. XML allows the encoding of arbitrary structures of hierarchical named values.
Security is a core component of the SDS and, where necessary,

communications are both encrypted and authenticated. SDS uses a Certificate Authority to provide a tool for authentication, and uses a Capability Manager to Maintain access control rights for users.

4. COMPARISON

Several service discovery protocols are proposed to facilitate dynamic cooperation among devices/services with minimal administration and human intervention. In order to support the requirement of dynamic service discovery, they should provide the means to announce services' presence to the network, to discover services in the neighborhood, and to access the services. Basically all the protocols we discussed address these aspects, but in different perspectives and emphases

Table 1 summarizes of features of major service discovery protocols

	SLP	JINI	SDS	INS
Developer	IETF	SUN	Berkeley Ninja	MIT LCS
Current Version	V2	V1.1	N/A	N/A
Main Entities	UA, SA and DA	Lookup Service, Service provider, and Client.	Clients, Services, and (SDS) Servers	Client, Service, INR Network
Service Description	Service URLs	Java class	XML	name-specifiers
Service Registration Lifetime	Soft state	Lease period	Soft state. Alive as long as announcement are made	Soft state. Alive as long as refreshments come
Security	Authentication	Java Based	Privacy and Authentication	
OS and Platform	Independent	Java	Independent	Independent
Late-bind concept	No	No	No	Yes
Network transport	Tcp/Ip	Independent	Independent	Independent
Serverless or Serverbased	Both	Serverbased	Serverbased	Serverbased

Table 1 summarizes the features of major service discovery protocols: SLP, JINI, SDS and INS. Among these, JINI and SLP come primarily from the industry; SDS and INS come from academics.

SLP:
SLP has a strong expressiveness of service definitions. Service URLs are organized into service types, and each type is associated with a service template that defines the required attributes. The functionality and expressiveness of this framework is almost an exact mapping onto the functionality of XML used by SDS. And the query correctness is more robust than JINI, which may cause query errors because of different class version.

Another obvious feature of SLP is scalability. Various features, such as the minimal use of multicast messages, scope concept, and multiple DAs support this. This has made SLP one of the most promising protocols for commercial use.

JINI:
Jini has a dependence on Java to enable all its promises. On the one hand, this concept makes JINI independent of the platform and operating system to run on. JINI uses java Remote Method Invocation (Java RMI) protocols to move program code around the network. This introduces the possibility to move device drivers to client applications, which is its main advantage over the non-Java based service discovery concepts. On the other hand, its requirement of Java Virtual Machine even in face of a JINI-proxy for a cluster of resource-poor devices has made portability to low-end devices a major concern.

INS:
INS has a flexible naming and resolution system for resource discovery, and it is well suited to dynamic network environments. INS uses a simple, expressive name language, late binding machinery that integrates resolution and routing for intentional anycast and multicast, soft-state name dissemination protocols for robustness, and a self-configuring

resolver network.

Unlike DNS, name propagation in INS resembles a routing protocol, tuned to perform rapid updates. The tight integration of naming and forwarding enables continued network connectivity in the face of service mobility, and the decentralized INS architecture and name discovery protocols enhance robustness.

SDS:

SDS seems to be unique in its well-defined security model. For example, SDS allows for private services to specify a list of people who can access information about the service; it protects against masquerading and holds components accountable for false information etc. SDS' security goals include 3 aspects: access control, authentication of all components and encrypted communication, which has made it the most secure protocol of the four.

The use of XML for service/device description distinguishes SDS as well. XML allows for powerful description of device capability, control command issued to the device, and events from it. The wise option has help it achieves the same, if not more than, strength in service definition and query as SLP.

5. CONCLITION

With the raising number of Internet services, automatic service discovery will be a very important feature in future networks. Several protocols have been proposed to address this problem, which resemble each other in the major concepts, as well as distinguish themselves by unique features. This paper attempts to take a look at key service discovery protocols such as: SLP, JINI, NIS and SDS, and compare them in system architecture, service representation, security issues etc. Their basic models are presented, and unique features are listed.

These service discovery protocols are vying with one another to be final winners. And they are also making efforts to integrate themselves with other protocols. For example, a JINI-to-SLP bridge developed by Sun is one of such efforts. However, It is unlikely that this diversity of protocols will continue for long. Mass production will bring us to a single standard. It is difficult to know which approaches will prevail. And it is quite likely that this will be another case where market share rather than technical merit will decide.

REFERENCES

[1] Czerwinski, Steven E., Zhao, Ben Y., Hodes, Todd D., Joseph, Anthony D., and Katz, Randy H, "An Architecture for a Secure Service Discovery Service," Mobicom'99, 1999. http://ninja.cs.berkeley.edu/dist/papers/sds-mobicom.pdf

[2] Fitzgerald, Steven, Foster, Ian, Kesselman, Carl, Laszewski, Gregor von, Smith, Warren, and Tuecke, Steven, "A Directory Service for Configuring High-Performance Distributed Computations," The 6th IEEE Symposium on High-Performance Distributed Computing, 1997. ftp://ftp.globus.org/pub/globus/papers/hpdc97-mds.pdf

[3] Goland, Yaron Y., Cai, Ting, Leach, Paul, Gu, Ye, and Albright, Shivaun, "Simple Service Discovery Protocol," IETF, Draft draft-cai-ssdp-v1-03, October 28 1999. http://search.ietf.org/internet-drafts/draft-cai-ssdp-v1-03.txt

[4] Guttman, Erik, "Service Location Protocol : Automatic Discovery of IP Network Services,"IEEE Internet Computing, vol. 3, no. 4, pp. 71-80, 1999. http://computer.org/internet/

[5] Mockapetris, P., "Domain Names-Implementation and Specification," IETF RFC 1035, October 1987. http://www.rfc-editor.org/rfc/rfc1035.txt

[6] "The Ninja Project," http://ninja.cs.berkeley.edu

[7] Perkins, C. and Guttman, E., "DHCP Options for Service Location Protocol," IETF RFC 2610, June 1999. http://www.rfc-editor.org/rfc/rfc2610.txt

[8] C. Perkins. DHCP Options for Service Location Protocol. draft-ietf-dhc-slp-02.txt, May 1997.

[9] Sun Microsystems. Jini technology architectural overview, white paper. http://www.sun.com/jini/whitepapers/architecture.html

[10] William Adjie-Winoto, Eliot Schwartz, Hari Balakrishnan, and Jeremy Lilley, The Design and Implementation of an Intentional.

[11] Naming System, (MIT Laboratory for Computer Science) 17th ACM Symposium on Operating Systems Principles (SOSP '99).

A New Information Search and Push System

Kong Yiqing, Fen Bin, Xu Wenbo
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu, 214036, China
Email: kongyiqing@263.net; fengb2001@263.net; xwb0121@china.com

ABSTRACT

In the paper, a new system for information search and push is introduced. The system will improve the quality of information search and ameliorates push technology used now. Thus the system can perform a better information service.

Keywords: Information Service, Search, Push.

1. INTRODUCTION

Information search and push technology has become the most important part in the browser technology. But there still have some problems when we use the technology.

If giving users information based on their requests, the situation will be improved efficiently. If gathering information from Internet, using data mining technology to find out the interests of users, then we can achieve the task of providing personalized information. By using push technology, the information provided could be updated when they have been changed.

The information gathering can be achieved by using search engines; the process of search results will improve accuracy of the answers. The information pushing can be achieved by establishing a push system. The system will push data to users through various channels. Users can choose the specific channel they like. The content in the channel can be pushed. To avoid too many requests to server, a push relay will be established. Then the network bandwidth is less occupied to solve the problem of network data transfer.

2. SYSTEM ARCHITECTURE

The information service system must achieve the task of picking up valuable materials. It should search, select and gather information from the web and could tell which information is the most suitable for using. To achieve this, an information searching system is established. When having the proper information gathered, an information mining system is established to deal with the information. Then an information pushing system is also established to push useful information. Thus, the information system can be divided into three main parts: the information searching system, the information mining system and the information pushing system.

Searching System

Classification of retrieval systems

There are several kinds of information retrieval systems existed on the web. Each has its own specific purpose for retrieval. When an Internet user enters his query, he is often confused about the different user interfaces and different functions of these retrieval systems. So it is quite necessary for us to make a classification of information retrieval systems on the Web.

The whole Internet information system can be divided into three layers [1], User Layer, Middle Layer and Web Layer. Fig.1 demonstrates Internet information system.

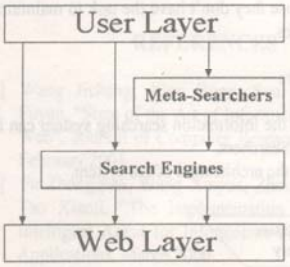


Fig.1 The classification of retrieval systems

Search Engines

As we all know, the web information fetching technology today basically depends on search engines. Every day, several of Robots, Spiders or Web Crawlers crawl over thousands of web pages to find out the latest contents. Then the newly found web pages are put into search engines' index databases. Thus, users can use these search engines to find out suitable pages through the index databases. Without these useful search engines, we can't even imagine how to use the powerful information supermarket. But the existed search engines have some problems.

1) The coverage of a single search engine is quite limited. There is not even a specific search engine that can involve most of the information on Internet.
2) The coverage of various search engines' index databases is quite different. The results given by different search engines are sometimes very different from each other.
3) The rule of results ranking is often not open to users, therefore we can't know which rule is most suitable for us.

Meta-Searches

The put forward of Meta-Searchers has overcome some of the shortcomings of search engines. The quality of information query can be improved by dealing with the results of search engines. Meta-Searchers themselves don't use Robots, Spiders or Web Crawlers to crawl over web pages, thus, they don't need to maintain the large index databases. At the same time, Meta-Searchers can provide a uniformed user interface for users to use. When a user enters his query, the query will be

changed into a query format that can be recognized by different search engines. The real query process is performed not by Meta-Searchers, but by search engines. And the results given by search engines must be processed in order to organize more precise and suitable answers before accommodate to users. By doing this, we can see that Meta-Searchers have several advantages over that of search engines.

1) The coverage of Meta-Searchers is wider than that of search engines, so some of the query results would not be left out.

2) The possibility of matching the request of users is being highly improved.

3) The uniformed user interface makes it easier for users to use and understand.

4) Developers can make the query results more precise and more suitable because they don't have the task to maintain the large index databases.

System architecture

The architecture of the information searching system can base on the use of Meta-Searchers.
Fig.2 demonstrates the architecture of the system.

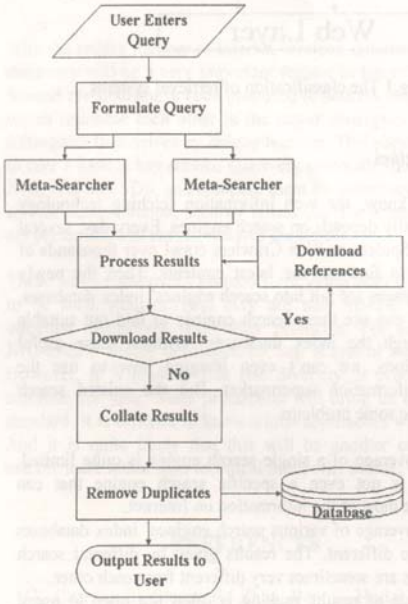


Fig.2 Searching system architecture

A user enters his keyword query for information searching. Formulate Query formulates various queries in order that various Meta-Searchers can understand the meaning of that. Then query process is dealt with by Meta-Searchers. The results of various Meta-Searchers must be dealt with. If the results need to be downloaded, then downloads that. Then we should collate results and remove duplicates. The processed answers can be put into database or just given to users.

Mining System

The information searching system has gathered so many we pages, therefore the information mining system has to find ou useful data by using data mining technology.
The interests of users can be found out through data mining by doing this, we will improve the accuracy of query.

System architecture

The architecture of the information mining system can b divided into several parts [2] [3].
User Interface: The interface between a user and the system.
Server: Serves users with information.
Information Management: Exchanges data between the Serve and database.
Log: Records web pages read by a user.
Learning Agent: Learns the character of the web pages tha have been read by a user.
Watching Agent: Checks weather the web pages that a user i interested in have been changed or not. I they have been changed, then update database.
Fig.3 demonstrates the architecture of the system.

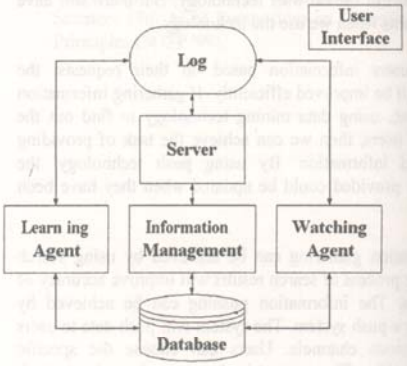


Fig.3 Mining system architecture

A user requests for proper information through User Interface Server receives the request, fetching web information from database through Information Management. When Serve responses the user's request, Information Management firs looks up the database. If the information involves in th database, then fetches the backup of that. Otherwise, fetches i from Internet through Meta-Searchers. Server creates a log fo every user, and saves that in Log. Learning Agent analyzes th log, finds out the interest of every user. Then it creates a use file that is helpful to find out which information is needed by the user. Watching Agent occasionally checks weather th web pages that a user is interested in have been changed or no If they have been changed, then updates database.

Pushing system

Push technology is the development of web browse technology. It is based on the demand of Intranet softwar distribution. Server wants to push the latest version o software to users other than rely on users' downloading by themselves. Then push technology became a uniformed ide for expressing pushing data to users without users' requests.
Software developers, like PointCast and BackWeb, hav developed some solutions for publishing information i

Intranet by using push technology. But these solutions are based on their different protocols and different working ways. The push technology used in IE and Netscape Communicator is based on HTTP.

The new way of information fetching relies on the concept of Channel. It changes the way of information publishing, software distribution and information exchanging in electronic commerce. But because push technology is just in its beginning, there are many problems. Now the application of push technology is used for pushing the same data to different users. If there were ten users, the same data would be pushed ten times. Because the same data is transferred, it creates the great problem of high bandwidth waste. When there are many users and the bandwidth is not wide enough, the push technology is usually impossible to use. In such situation, we also can't push video or audio data through channels.

The concept of channel relay

In order to solve the high bandwidth consuming that limits the development of push technology and improves the efficiency of channel message distribution, the concept of channel relay has been proposed [4].

The basic idea of channel relay is from the information fetching way of Publishing and Subscribing. The traditional information fetching way in Internet is Request and Reply. But it would be difficult to achieve when there are too many users. The new information fetching way of Publishing and Subscribing can be applied in many circumstances.

The work way of channel relay

The channel relay is the application of Publishing and Subscribing in information channel technology. The push server will push information to Intranet users. The specification of each channel will be defined, including the update frequency. According to the frequency, server pushes the information on the web to PCs. This costs most of the bandwidth when there are many users. The efficiency is low, and it can't push video, audio data or database information. The reason for the waste of bandwidth is that the same information is pushed to every user. In order to avoid transferring the same data on low-speed Internet, a channel relay between information provider and receiver will be established.

Fig.4 demonstrates the architecture of the system.

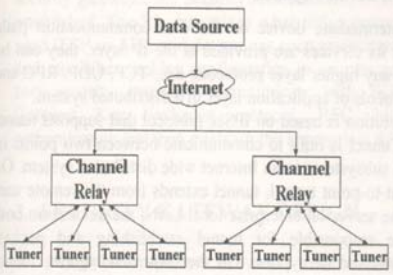


Fig.4 Pushing system architecture

The channel relay downloads information from the provider. The information will be received through high-speed Intranet. Information channel relay push is much better than pure push technology. The situation of high cost of bandwidth is avoided. If users increase, more channel relays will be established to improve situation. The establishment of a channel relay could be used in information publishing, e-business and updating of software, etc.

3. CONCLUSIONS

The establishment of a new system for information search and push can be applied in many fields.

1) ISPs and ICPs will push various data to users who need them.
2) An enterprise will publish information about itself to different users who are interested in its products.
3) Internet users will receive the latest news that they concern about.

REFERENCES

[1] Wang Jicheng, Xiao Rong, Sun Zhengxing, Zhang Fuyan, "State of the Art of Information Retrieval on the Web", Journal of Computer Research & Development, February 2001.
[2] Fu Zhongqian, Wang Xinyue, Zhou Peiling, Peng Hu, Tao Xiaoli, "The Implementation of a Personalized Intelligent Agent for Information Filtering", Computer Applications, March 2000.
[3] Chen Ning, Zhou Longxiang, "Application of Data Mining in Internet", Computer Science, July 1999.
[4] Mao Weihua, Guo Zonggui, "Channel relay: A way of push technology used on Internet and Intranet", Computer Development and Application, April 1999.

The Research and Design on Secure Tunnels and Security Authentication of Distributed Computing and Application

Lu Jiande
Department of Computer, Suzhou University
Suzhou, Jiangsu 215006, P.R. China
E-mail: lujiande@public1.sz.js.cn

ABSTRACT

A very important issue of Internet wide distributed computing and application is the distributed system security. On the basis of thorough analysis of secure tunnels and security authentication, this paper has discussed the methods of enhancing the security between IP nodes of an Internet wide distributed system and detailed design of security gateway and key exchange.

Keyword: Tunnel, Authentication, IPSec, IKE, SA

1. INTRODUCTION

When we make distributed computing and application on Internet, Internet security becomes the very important problem to an Internet wide distributed system. Internet is originally designed on the basis of trust between users. Because Internet information transmission takes forms in clear text, hackers may easily intercept and capture data packets on the Internet. The distributed system users can't be reassured with Internet as their secure information transmission media, commercial users also can't be reassured with Internet do their electronic commerce and trading.

In an Internet wide distributed system, we can design a security gateway enhancing the security between distributed sites. With this security gateway resolution, the distributed system users

can make their virtual private connection using existent inexpensive Internet public network. When we design security gateway, we can consider using IPSec protocol with Encapsulating Security Payload protocol ESP for confidentiality (encryption) and Authentication Header protocol AH for integrity and data origin authentication.

IPSec security services are provided at the IP layer, they can be used by any higher layer protocol, e.g., TCP, UDP, RPC, distributed file system and distributed database. Prior to IPSe processing the system need to build appropriate SAs (Securit Associations) using IKE protocol. This paper will discuss ou resolution, a security gateway resolution, based on IPSec an IKE protocols and its design and implementation on Linux.

2. SECURITY GATEWAY RESOLUTIO BASED ON IPSEC AND IKE

In RFC 2401--2409, IETF proposed series protocols IPSe enhancing security between IP nodes. The purpose of IPSec i to provide security services at the IP layer, its securit protocols mainly consist of ESP protocol and AH protocol ESP refers to payload encapsulation and encryption, and AH refers to integrity and data origin authentication. Select one o them or both. IPSec protocol is performed between IP end-to- end hosts or security gateways to protect IP data packet transmission. IPSec can be used protecting IP packets on on or multiple path.

By implementing IPSec protocol on router, firewall or multi- homed PC, we can build a security gateway. Security gateway

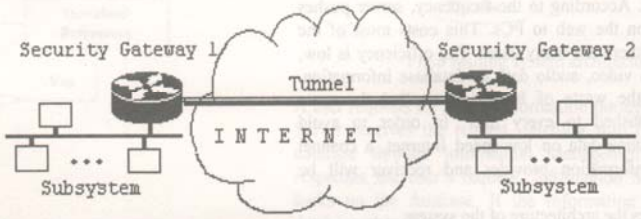


Fig-1 Secure Internet wide distributed system

is an intermediate device on the data communication path. Because its services are provided at the IP layer, they can be used by any higher layer protocols, e.g., TCP, UDP, RPC and any protocols of application layer in a distributed system.

The resolution is based on IPSec protocol that supports tunnel mode. Tunnel is built to communicate between two points or two end subsystems in an Internet wide distributed system. On the point-to-point tunnel, tunnel extends from the remote user PC to the server on enterprise local LAN, the devices on both side are responsible for tunnel establishing and packets encryption and decryption. On the end-to-end tunnel, tunnel connects two local LANs and terminates at each security gateway of the local LAN. When multiple paths (e.g., from the different security gateways) exist to the destination behind the security gateway, multiple tunnels are needed Internet wide distributed system uses tunnel communication between security gateways. Users can be reassured with Internet sending their IP packets encrypted and encapsulated by ESP,

though someone may intercept and capture that data packets, the packets are useless to them because they don't possess that associated key. Users can also be reassured with Internet sending their IP packets that are data origin authenticated and integrity checked by AH. In this resolution, the security gateways, the subsystems behind security gateways and the tunnel through Internet constitute a secure Internet wide distributed system as illustrated in Fig-1.

The plain IP packets are transmitted on subsystem. When arrive at the security gateway, they are processed by gateway software, then the transformed IP packets are transmitted on the Internet (tunnel) between security gateway 1 and 2, the protocol header in a packet may look like the following:

(1) When using ESP protocol for encryption of data:
[New_IP_header][ESP_header][Original_IP_header][Upper_layer_protocol_header]
or:
(2) When using AH protocol for authentication of data:
[New_IP_header][AH_header][Original_IP_header]
[Upper_layer_protocol_header]

In (1), the fields after ESP header are all encrypted. In (2), all fields are authenticated. In (1) or (2), outer [New_IP_header] includes source and destination security gateway IP address; inner [Original_IP_header] is the original IP header before arriving at the security gateway and includes the packet's ultimate destination IP address. ESP header or AH header is located after outer new IP header and before inner original IP header. The protocol header immediately preceding the ESP header will contain the value 50 in its "Protocol" field. The value 51 will do in the case of AH.

To implement IPSec security gateway, put IPSec processing code into the network layer of the host operating system kernel. This processing makes combination of IP code and IPSec code more tight and succinct but need analyzing and accessing host's source kernel code. Microsoft Windows NT kernel code is not open to the outside and Linux, a very powerful and popular operating system, is a best choice now. We need analyzing Linux kernel source code and inserting IPSec protocol and other processing code into the appropriate place of the kernel, then recompiling it and generating the new kernel that has security gateway functions. We have developed such prototype on RedHat Linux 5.1(kernel v2.0.34).

Before two hosts make their secure communication through security gateways, the Security Associations (SAs) need to be established. The SA determines what IPSec protocol is to be used (ESP or AH), and determines encryption algorithm, key, and etc. When making configuration to security gateway, the two security gateways communicate with IKE protocol (Internet Key Exchange protocol, RFC 2409), make authentication and then establish the appropriate SA item.

3. SECURITY GATEWAY DESIGN

Security gateway opens or terminates a tunnel as needed. The IKE authenticates the establishing tunnel request and forms the SA. According to the established SA, the security gateway opens the tunnel. The ESP or AH packets transmitted on the tunnel use the IP addresses of the tunnel opener and the tunnel terminator (i.e., the IP addresses of two security gateways) to screen the addresses of the original source and destination.

3.1 Using IKE to Establish SA

When establishing SA (Security Association) through IKE protocol, system need negotiate between two security gateways and make authentication to them. IKE uses ISAKMP language to describe key exchange. On each security gateway, system initiates an IKE server daemon, running it in the background and the IKE server keeps listening on UDP port 500. The security gateway that makes request runs a client program to communicate with the destination security gateway UDP port 500.

If we call the server process the "Responder" and the client program the "Initiator", after negotiation, a unidirectional SA will be established between two security gateways, the communication process is shown as the Fig-2.

3.2 Security Policy Database (SPD) Design

The security policy database (SPD) is built on security gateway. It stores filtering policies, protocols and associated security services, and etc. These policies determine how to process the inbound or outbound IP packet. The security gateway looks up the policy in the SPD to see if it matches the inbound or outbound packet. According to the match result, it can determine:

(1) discard,
or (2) bypass,
or (3) IPSec processing.

When the matched policy denies the packet passing, simply discard it. When the matched policy permits the packet passing without IPSec processing, bypass the IPSec processing. When the matched policy indicates it needs IPSec processing, then the packet is passed to IPSec module. If there is no matched policy in the SPD, the packet must also be discarded.

According to this design, in the Linux, the security policy file /etc/vpn.conf is configured to include the SPD permission, denial, and bypass policies. Each policy statement format is as the following:

IfPermit = Protocol | FromNet | FromMask | FromPortBegin | FromPortEnd | ToNet | ToMask | ToPortBegin | ToPortEnd | PtrToSendSA

The 'IfPermit' can take the value permission, denial or bypass. The 'Protocol' is the packet protocol. 'FromNet' and 'FromMask' indicate the packet source IP address and the netmask respectively. The 'FromPortBegin' and 'FromPortEnd' are defined as the minimum and maximum port number allowed or denied on source side TCP/UDP port. 'ToNet' and 'ToMask' indicate the packet destination IP address and netmask respectively. The 'ToPortBegin' and 'ToPortEnd' are defined as the minimum and maximum port number allowed or denied on destination side TCP/UDP port. In the last, 'PtrToSendSA' indicates SA pointer when sending outbound packet to the tunnel. Every time Linux starts and performs security gateway initialization, the kernel reads in every policy in the SPD and stores them into a data structure in the memory for the future use.

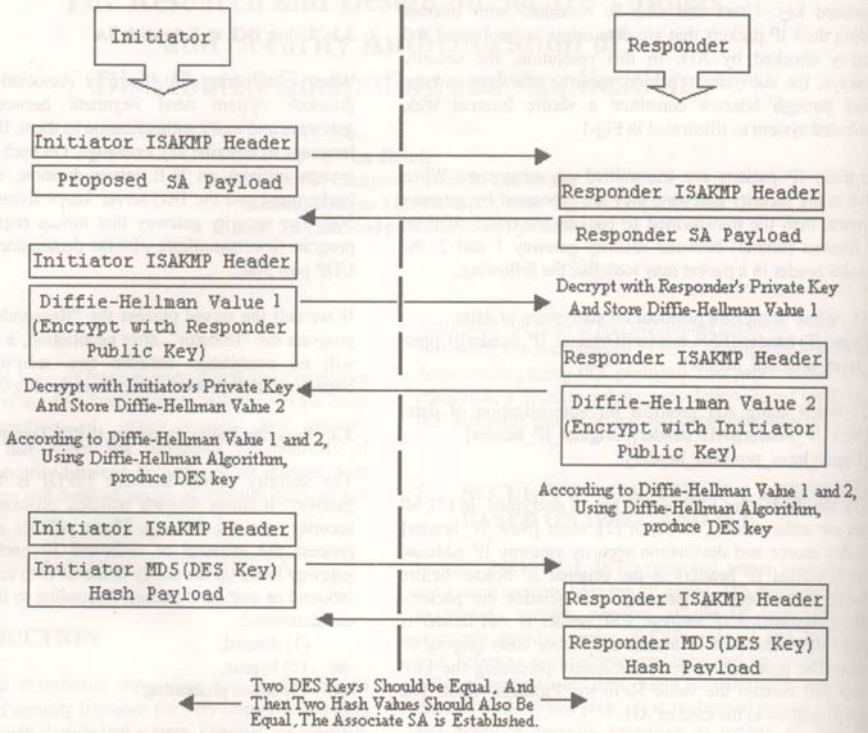


Fig-2 Communication Using ISAKMP and IKE Protocol

3.3 Security Association Database (SAD) Design

System employs a data structure, called security association (SA), to describe a simplex "connection" that affords security services to the traffic carried by it. Security services are afforded to a SA by the use of AH, or ESP, but not both. If both AH and ESP protection are applied to a traffic stream, then two (or more) SAs are created to afford protection to this traffic stream. To secure typical, bi-directional communication between two security gateways, two Security Associations (one in each direction) are required. To support multi-tunnel scheme, when multiple tunnels from various sources terminate at a security gateway, every traffic stream on the tunnels needs to be described using a special SA.

A security association is uniquely identified by a triple consisting of a Security Parameter Index (SPI), an IP destination address, and a security protocol (AH or ESP) identifier. Except these three parameters, every SA has also parameters sequence number, ESP encryption algorithm, AH authentication algorithm, key, initial vector and etc.
In our Linux kernel design, every security gateway has been devised to create a security association database (SAD) file /etc/vpn.sad, which is configured and stored to include several SAs. ESP or AH is applied to an outbound packet only after IPsec code determines that the packet is associated with a SA that calls for ESP or AH processing.

When sending a packet onto some tunnel using IPSec, at first, look up the SA pointer 'PtrToSendSA' in the matched permission policy of the SPD, and then, according to pointer, find the appropriate SA in the SAD.

When receiving a packet on some tunnel using IPS according to the destination address, security protocol (AH ESP) and Security Parameter Index (SPI) in the recei packet, look up the appropriate SA in the SAD, then m appropriate IPSec processing by that SA's security descr parameters.

3.4 ESP and AH Header Design

ESP header and packet, AH header and packet have the form as shown in Fig-3.

The function of SPI is: when multiple tunnels terminate a security gateway, if the IP destination and the secur protocol in the IP packets came from the different tunnels a the same, system can still recognize the different tunnel's S by the SPI value.

After establishing tunnel, the sequence number of every ne ESP or AH packet transmitted on the tunnel increases by (the first number is 1), different to each other. The function Sequence Number is to prevent the third parties from attackin by re-using some sequence numbers that have already bee used.

The ESP payload in the fig-3 is the encrypted original IP

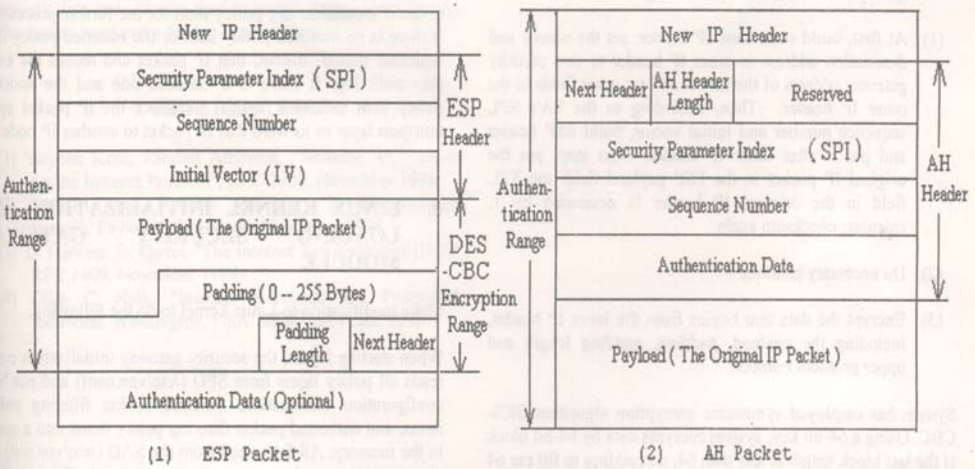


Fig-3 ESP and AH packet

packet. Normally, the encryption algorithm is DES-CBC, and the algorithm needs an initial vector, which is indicated by the Initial Vector field.

DES-CBC algorithm encrypts data flow by 64-bit block. When the original IP packet bit length divided by 64 is not an integer, the processing needs padding. Padding length field indicates the padding length.

In ESP packet, authentication data field is optional, which is the integrity value calculated to ESP packet; the receiver would use this field for authentication. In AH packet, the authentication data field is the whole packet's integrity value.

The Next Header field is 4 to indicate IP-in-IP

3.5 Analysis of Insertion Point in Linux Kernel

The key design of the security gateway is implementing packet filtering and data encryption. Through the TCP/IP protocol stack analysis to the Linux kernel, it is found that the function ip_rcv() in the kernel is used for receiving the IP packets from MAC, and according to its destination address system can determine if to pass to the upper layer for the next processing or to forward to another IP node.

When forwarding, the function ip_forward() is employed and this function will call function dev_queue_xmit() to pass IP packet to a queue for transmission.

On the other hand, the data sent to the other node from the upper layer go down through the TCP/IP protocol stack and finally would also call dev_queue_xmit() to pass to MAC NIC driver. In addition, the other IP packets sent from this machine would go through the function ip_queue_xmit() and ip_build_xmit(), and finally would also call dev_queue_xmit() to pass to MAC NIC driver.

According to above analysis, we can consider to insert the key code for security gateway ESP processing into the kernel function ip_rcv(), ip_queue_xmit() and ip_build_xmit(), so as to intercept all traffic stream we concerned.

On the ICMP processing, in the Linux kernel the ICMP packet sent is passed directly to dev_queue_xmit(), so in function dev_queue_xmit() system also needs to do security processing to the ICMP packet.

Finally, there is something that still needs us to consider. Because IPSec processing need add new IP header, ESP or AH header, ESP tail and necessary padding, this requires some memory space reservation. We need to find the memory allocation function call in the packet-processing module and make modification to it.

In Linux kernel, there are some indirect recursive function-calls. For example, in function ip_queue_xmit(), if packet length is greater than MTU, it would call function ip_fragment() for fragmentation. But after fragmentation, in the function ip_fragment(), it would call ip_queue_xmit() again, also in function ip_forward(), system would call ip_fragment() for fragmentation. We need define a flag, making ESP (or AH) processing do only once.

4. LINUX OUTBOUND PACKET PROCESSING

In Linux kernel, the IPSec processing code inserted in function ip_forward() and dev_queue_xmit() realizes the following function:

According to IP header, look up in SPD to see if it matches some policy item line. If there is no matched policy item or the matched item is configured 'denial', discard the IP packet and record it in audit log. If the matched policy item is configured 'bypass', introduce the IP packet to go through the original logic in dev_queue_xmit(). If the matched policy item is configured 'permit', then get the pointer PtrToSendSA in the matched policy item and find associate SA in SAD. If that pointer is NULL, then call tunnel authentication and establishing module, build a new SA item in the SAD for new established tunnel, and modify the PtrToSendSA in the SPD. Then according to found SA or new established SA, do IPSec processing on the basis of that SA's security described parameters.

When system makes ESP send processing:

(1) At first, build outer new IP header: set the source and destination address in outer IP header to two security gateway address of the tunnel and set other fields in the outer IP header. Then, according to the SA's SPI, sequence number and initial vector, build ESP header and put it after outer IP header. Next step, put the original IP packet in the ESP payload field, the TTL field in the original IP header is decreased by 1, calculate checksum again.

(2) Do necessary padding.

(3) Encrypt the data that begins from the inner IP header, including the payload, padding, padding length and upper protocol number.

System has employed symmetric encryption algorithm DES-CBC. Using a 64-bit key, system encrypts data by 64-bit block. If the last block length is less than 64, do padding to fill the 64 bits and then do encryption.

The system will make the similar AH send processing, but do authentication and no padding. If the new IP packet length after processing is greater than MTU on the outbound link, divide the packet to fragments of smaller size.

5. LINUX INBOUND PACKET PROCESSING

In Linux kernel, the IPSec processing code inserted in function ip_rcv() realizes the following function:

At first, the processing code checks IP packet to see if it is an IP fragment. If it is, reassembling is performed prior to the next processing.

Then, if the "Protocol" field in the outer IP header is 50, do ESP receiving processing; if the "Protocol" field in the outer IP header is 51, do AH receiving processing.

According to the destination address, IPSec protocol type and SPI in the received IP packet, look up in the SAD to see if it matches any SA. If there is no matched SA, discard the received IP packet and record the event into audit log. If there is, do IPSec processing on the basis of that SA's security described parameters.

When system makes ESP receive processing:

(1) According to the encryption algorithm and key involved in that SA, decrypt the data, which begins from the inner IP header, including the payload, padding, padding length and upper protocol number field.

(2) Process the padding field. Check the decrypted padding bitmap to see if any error occurs and deletes the padding field.

(3) Process the decrypted payload to restore the original IP packet.

The system will make the similar AH receive processing but do authentication and no padding.

According to the restored IP packet header, look up in the SPD to see if it matches any policy item for the further processing. If there is no matched policy item or the matched policy item indicates 'denial', discard that IP packet and record the event into audit log. If there is a matched one and the matched policy item indicates 'permit', introduce the IP packet up to transport layer or forward that IP packet to another IP node.

6. LINUX KERNEL INITIALIZATION AND LOADING SECURITY GATEWAY MODULE

Make modification to Linux kernel to do the following:

When starting Linux, the security gateway initialization codes reads all policy items from SPD (/etc/vpn.conf) and put NIC configuration information, inbound packet filtering policy items, and outbound packet filtering policy items into a queue in the memory. All SA's read from the SAD (/etc/vpn.sad) are also put into another queue in the memory. This memory cache is required for processing speed and efficiency. Later on, the security gateway software will look up in these queues for the search purpose when it does IP packets receiving and sending processing.

7. ENHANCED SECURITY CHECK

For enhanced security, we can also include filter policies and programming logic in the security gateway software to prevent hackers from attack. When system intercepts the traffic stream, the security gateway does the following:

If the received IP packet involves IP source route option, simply discard it. Some hackers would use the IP source route option to send an IP packet that appears to be sent from inside.

Sometimes, hackers would make IP address spoofing and masquerading. When system checks the source IP address in the received IP packets, if the IP packet is from the external interface but the source address in the header is the internal IP, also discard that packet.

To avoid SYN flood attack, system tracks received IP packets in which the SYN flag is set, security logic compares IP address, port number and other header information to see if the attacker continuously sends the connection requests. If do so, discard that IP packets which arrive at the external interface.

To avoid replayed packet attack, security code checks the ESP or AH packet sequence number field to prevent the repeated sequence number ESP or AH packet from coming in.

8. CONCLUSION

Through application of our designed security gateway based on IPSec protocol, we have monitored and captured IP packets that are transferred between the tunnel opener and tunnel terminator. It is found that the DES-CBC encrypted data flow on the tunnel is good in confidentiality, AH authenticated data is kept integrity, and the system also has filter-type firewall function and anti-attack's capacity. The system has reached the prototype's design goal. But due to the additional data encryption and authentication, the load of security gateway

increases, the traffic also increases by 30% to 40%, and the network data transmission time becomes longer than previous. The system need make further optimization.

REFERENCES

[1] Stephen Kent, Randall Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998
[2] Stephen Kent, Randall Atkinson, "IP Encapsulating Security Payload", RFC 2406, November 1998
[3] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998
[4] Dilip C. Naik, "Internet Standards and Protocols", Redmond, Washington, USA: Microsoft Press, 1998

A New Quick Public Key Crypto-System Based On the Difficulty of Factoring Very Large Numbers

Xiao Youan Li Layuan
College of Information Engineering
Wuhan University of Technology
Wuhan, Hubei, 430063, P. R. China
Email: youan@126.com

ABSTRACT

This paper introduces the new Quick Public Key Crypto System (QPKCS) based on the difficulty of factoring very large numbers. QPKCS is a public key crypto-system designed to provide a stronger & faster alternative to the well-known RSA algorithm.

Although the key generation process of the new system may be slower than that of a well-known existing system, the new system is much faster in the encryption and decryption process. The most notable advantage of the new system is its security feature. Since the key is different for different message blocks, it can reduce the risk of Common Value Detection attack.

Some suggests to fast the key generation process by the Parallel Distributed Computing is also discussed.

Keywords: Public Key Algorithms, Factoring Numbers, Euclidean Algorithm, Euler Function, BK Method

1. INTRODUCTION

In 1976, Whitfield Diffie, Martin Hellman and Ralph Merkle, developed the theory what is known as public key cryptography. It's an important revolution in cryptography. Their contribution to cryptography was the notion that keys could come in pairs—an encryption key and a decryption key—and that it could be infeasible to generate one key from the other [1].

That is say, the Public Key cryptography is an encryption/decryption method where two keys are generated, a public key and a private key. Data encrypted with the public key can only be decrypted using the private key. Public keys do not require secrecy. They can be safely exchanged through unsecured channels without compromising system integrity. Only disclosure of the private key is a compromise, and only one person's communications are vulnerable.

The first Public-Key Encryption Algorithm was the knapsack algorithm developed by Ralph Merkle and Martin Hellman. However, the knapsack algorithm was later found to be insecure.

Soon after Merkle's knapsack algorithm came the first full-fledged public- key algorithm, one that works well for both encryption and digital signatures: RSA. It's developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Of all the Public-Key Encryption Algorithm proposed over the years, RSA is by far the easiest to understand and implement. And it has since withstood years of extensive cryptanalysis. So it is thought as both secure and practical, and is the most popular Public-Key Encryption Algorithm today. But it is very slow. They encrypt and decrypt data much more slowly than common symmetric algorithms; usually that's too slow to support bulk data encryption.

To solve this problem, we introduce a new Quick Public Key Crypto-System (QPKCS) based on the difficulty of factoring very large numbers (which is same as RSA). And it is designed to provide a stronger & faster alternative to the well-known RSA algorithm.

This paper is organized as follows. Section 2 describes the well-known RSA Public-Key Encryption Algorithm. Section 3 analyses the problems of the RSA algorithm. To solve these problems, a new Quick Public Key Crypto System is introduced and discussed in Section 4 and 5. Section 6 compares the new QPKCS algorithm with the RSA algorithm. Section 7 discusses some idea to improve the QPKCS algorithm with Distributed BK algorithm. Finally, in Section 8, we provide a conclusion.

2. RSA PUBLIC-KEY ENCRYPTION ALGORITHM [3][4]

RSA gets its security from the difficulty of factoring large numbers. The public and private keys are functions of a pair of large (100 to 200 digits or even larger) prime numbers. Recovering the plaintext from the public key and the cipher text is conjectured to be equivalent to factoring the product of the two primes.

To generate the two keys, choose two random large prime numbers, p and q. For maximum security, choose p and q of equal length. Compute the product, and the Euler Totient Function $\varphi(r)$:

$$r = pq \tag{2.1}$$

$$\varphi(r)=(p-1)(q-1) \tag{2.2}$$

Then randomly choose the encryption key, E, such that E and $\varphi(r)$ are relatively prime. Finally, use the extended Euclidean algorithm to compute the decryption key, D, such that

[1] The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

$$D = E^{-1} \bmod \varphi(r) \tag{2.4}$$

Note that D and r are also relatively prime. The numbers E and r are the public key; the number D is the private key. The two primes, p and q, are no longer needed. They should be discarded, but never revealed.

To encrypt a message M, first divide it into numerical blocks smaller than r (with binary data, choose the largest power of 2 less than r). That is, if both p and q are 100-digit primes, then r will have just under 200 digits and each message block, Mi, should be just under 200 digits long. (If you need to encrypt a fixed number of blocks, you can pad them with a few zeros on the left to ensure that they will always be less than r.) The encrypted message, C, will be made up of similarly sized message blocks, Ci, of about the same length.

The encryption formula is simply:

$$C_i = M_i^E \bmod r \tag{2.5}$$

To decrypt a message, take each encrypted block Ci and compute

$$M_i = C_i^D \bmod r \tag{2.6}$$

3. PROBLEMS IN THE RSA ALGORITHM

3.1 Speed of RSA[5]

One of the most important problems of RSA Algorithm is that it's too slow. Calculating powers can be very cumbersome, particularly when the exponent is 128 bits wide or larger. This problem causes the RSA algorithm to be very slow, about more than 1000 times slower then DES. The fastest VLSI hardware implementation for RSA with a 512-bit modulus would struggle to reach a throughput of 64 kilobits per second [2]. These numbers may change slightly as technology changes, but RSA will never approach the speed of symmetric algorithms.

3.2 Security of RSA

RSA algorithm has since withstood years of extensive cryptanalysis. The security of RSA depends wholly on the problem of factoring large numbers. It is conjectured that the security of RSA depends on the problem of factoring large numbers. It has never been mathematically proven that you need to factor N to calculate M from C and E. But, an entirely different way to crypt analyze RSA hasn't been found too.

Although the cryptanalysis neither proved nor disproved the security of RSA, it does suggest a confidence level in the algorithm.

Today, many attacks work against the implementation of RSA. These are not attacks against the basic algorithm, but against the protocol. Known attacks are listed as bellows:

- Chosen Cipher Text Attack;
- Common Modulus Attack;
- Low Encryption Exponent Attack;
- Low Decryption Exponent Attack;

As we enter the new millennium and computer computation speeds become faster, another problem with RSA becomes apparent, common value detection. Since RSA provide no ability to alter the initial keys, data is always encrypted with the same value. Hence x will always equal y in any instance x is encrypted.

For example, use the key pairs E = 4561 / D = 3649 / r = 11303 to encrypt the message "letter", we can get the crypto text as "10CB 07B2 0251 0251 07B2 0979". Then we can find that the encrypted of each char "e" is always "07B2", and "t" is "0251"!

Because the cryptanalyst has the right to access to the public key, so he can always choose any message to encrypt. This means that a cryptanalyst, given C = EK(P), can guess the value of P and easily check his guess. This is a serious problem if the number of possible plaintext messages is small enough to allow exhaustive search, which is common in the native language. To solve this problem, we must make identical plaintext messages encrypt to different cipher text messages. One of the methods is padding messages with a string of random bits, which is called as Probabilistic Encryption, but it's impractical.

4. THE QPKCS ALGORITHM

To solve the above main problems found in the RSA system, we have attempted to develop a new Quick Public Key Crypto System, which is called as QPKCS. It based on the difficulty of factoring very large numbers as RSA algorithm.

The simplest way to understand the QPKCS algorithm is to break it up into small digestible pieces. For each message block Mi, Mi must be smaller than r. In this algorithm, we attempt to alter the initial keys in the origin RSA algorithm, and introduce a new value called Message Index to control the key pairs.

Below is the algorithm its self. D is the decryption key, DI is the decryption index, E is the encryption key, EI is the encryption index, M is the message, MI is the message index, C is the cipher message and CI is the cipher text index.

4.1 Key Generation

To generate the two keys, choose two random large prime numbers, p and q. For maximum security, choose p and q of equal length. Compute the product, and the Euler Totient Function φ(r):

$$r = pq \tag{4.1}$$

$$\varphi(r) = (p - 1)(q - 1) \tag{4.2}$$

Then randomly choose the encryption key, E, such that E and φ(r) are relatively prime, then, choose a random large number, which fit the condition gcd(a, n) = 1. Finally, use the extended Euclidean algorithm to compute the decryption key, D, such that

$$E * D \equiv 1 \bmod (a^{\varphi(r)} - 1) \tag{4.3}$$

Now we can get the public and private keys. The public key is the pair of numbers (E, r), and the private key is the pair of numbers (D, r). The three numbers, a, p and q, are no longer

Now we can get the public and private keys. The public key is the pair of numbers (E, r), and the private key is the pair of numbers (D, r). The three numbers, a, p and q, are no longer needed. They should be discarded, but never revealed.

4.2 Session Initialization

Because the QPKCS algorithm introduces the MI, EI and DI, we should initialize the three variant at first:

The MI value is the location index of current message block in the source plain text.

Choose a random number as the initialization value of encryption index EI, and it should be less than r.

Compute the initialization value of decryption index DI, such that $EI + DI = r$.

Now, we have got the initialization value of EI and DI.

Note that the initial EI value should be known by any party commencing a session with E or D. Any EI value after the initial one should be kept secret. All DI values should be kept secret. We think that 1 is a good initial value for EI and may be used as a standard.

4.3 Encryption/Decryption

To encrypt a plain message M, first divide it into numerical blocks smaller than r as RSA algorithm. Each plain message block is called as Mi, and the index of every plain message block is MI. And each clipper message block is called as Ci, and the index of each clipper message block is CI, which is always equal to the value of MI.

The encryption formula is:

$$C_i = ((M_i + EI) * E) \bmod r \tag{4.4}$$

$$EI = (EI * (MI * M_i + 1)) \bmod r \tag{4.5}$$

$$MI = (MI + 1) \bmod r \tag{4.6}$$

To decrypt a message, take each encrypted block Ci and compute:

$$M_i = ((C_i * D) + DI)) \bmod r \tag{4.7}$$

$$DI = (DI * (CI * M_i + 1)) \bmod r \tag{4.8}$$

$$CI = (CI + 1) \bmod r \tag{4.9}$$

The Decryption Verification formula is:

$$EI + DI = r \tag{4.10}$$

After encrypt or decrypt a plain/clipper message block, we must update the EI and DI value. After the decryption, we should use the Decryption Verification formula to verify the decryption process.

5. PROVING

First, check the Decryption Verification Formula: $EI + DI = r$

1) $EI_0 + DI_0 = r$;
2) suggest that $EI_{n-1} + DI_{n-1} = r$;
3) for n, because CI = MI,

$$\begin{aligned} & EI_n + DI_n \\ &= (EI_{n-1} * (MI * M_i + 1) + DI_{n-1} * (CI * M_i + 1)) \bmod r \\ &= (EI_{n-1} + DI_{n-1}) * MI * M_i) \bmod r + (EI_{n-1} + DI_{n-1}) \\ &= r * MI * M_i \bmod r + r \\ &= r \end{aligned}$$

So, for each pairs of EI and DI, the Decryption Verification Formula is always right.

To prove the QPKCS algorithm, we should compute the $DK(EK(M_i))$ for each plain message block M_i. As below:

$$\begin{aligned} & DK(EK(M_i)) = ((EK(M_i) * D) + DI) \bmod r \\ &= ((((M_i + EI) * E) \bmod r) * D + DI) \bmod r \\ &= (((M_i + EI) * E) * D + DI) \bmod r \\ &= (M_i * E * D + EI * E * D + DI) \bmod r \\ &= (M_i * E * D + EI * E * D + r - EI) \bmod r \\ &= (M_i * E * D + EI * (E * D - 1) \bmod r \\ &= (M_i + (M_i + EI) * (E * D - 1)) \bmod r \end{aligned}$$

Because $E * D \equiv 1 \bmod (a^{\varphi(r)} - 1)$, so $(E * D - 1) \bmod (a^{\varphi(r)} - 1) = 0$. According to Euler's generalization of Fermat's little theorem, with $\gcd(a, r) = 1$, then

$$a^{\varphi(r)} \bmod r = 1,$$

that is say, $(a^{\varphi(r)} - 1) \bmod r = 1$,

So, we can get that

$$(E * D - 1) \bmod r = 0$$

Therefore,

$$\begin{aligned} & DK(EK(Mi)) \\ &= (Mi + (Mi + EI) * (E * D - 1)) \bmod r \\ &= Mi \end{aligned}$$

So, the QPKCS Algorithm has been proved.

6. COMPARED WITH THE RSA ALGORITHM

6.1 Key Generation

Because the QPKCS Algorithm uses the $a^{\varphi(r)}$ to calculate D, while the RSA Algorithm use the $\varphi(r)$. So the speed of key generation is slower than that of RSA.

6.2 Encryption/Decryption

The QPKCS Algorithm uses only three additions and multiplications to encrypt / decrypt the message block, while

RSA, and is what we expected.

6.3 Security of Algorithm

With the introduction of MI/CI, EI and DI, the key is varying for each different message block. The same message block isn't encrypted with the same clipper block. So the common value diction attack can't be affect again.

That is to say; it would be safer than RSA in Common Value Diction Attack.

7. ALGORITHM IMPROVING

The new QPKCS algorithm is better than the RSA algorithm except the speed of key generation.

For both QPKCS algorithm and RSA algorithm, the process of key generation is the search of the decryption key D. Because the complexity of searching key is same for both algorithms, the main reason of slower key generation speed is the QPKCS Algorithm uses $a^{\varphi(r)}$ while RSA Algorithm uses $\varphi(r)$.

To improve the performance of key generation in QPKCS, we can use the Distributed BK algorithm [5] to calculate the $a^{\varphi(r)}$, and can cut the complexity from $O(a^n)$ down to $O(n)$.

8. CONCLUSIONS

RSA algorithm is one of the most popular public key encryption algorithms, and can be used as both secure and practical. But it is too slow to support bulk data encryption. Also, the RSA algorithm can't withstand the common value detection attack when the block size isn't large enough

The new QPKCS algorithm is an alternative to the well-known RSA algorithm. It has a faster encryption / decryption speed and better security than the RSA algorithm. So it can be used in every field that RSA algorithm is used, include Digital encryption / decryption and Digital Signature Algorithms.

Due to the speed of key generation, we plan to improve the algorithm key generation and reduce the complexity of it.

REFERENCES

[1] W. Diffie and M.E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, v. IT-22, n.6 Nov 1976, pp.644-654

[2] E.F.Brickell, "Survey of Hardware Implementations of RSA", Advances in Cryptology – CRYPTO'89 Proceedings, Springer-Verlag, 1990, pp. 368-370.

[3] R.L. Rivest, A. Shamir, and L.M. Adleman, "On Digital Signatures and Public Key Cryptosystems", MIT Laboratory for Computer Science, Technical Report, MIT/LCS/TR-212. Jan 1979

[4] R.L. Rivest, A. Shamir, and L.M. Adleman, "Cryptographic Communications System and Method", U.S. Patent #4,405,829,20 Sep 1983

[5] B.Schneier, "Applied Cryptography Second Edition: protocols, algorithms, and source codes in C", John Wiley & Sons Inc., Jan 1996

[6] Martín Abadi, Phillip Rogaway: "Reconciling Two Views of Cryptography (The Computational Soundness of Formal Encryption)", Proceedings of the First IFIP International Conference on Theoretical Computer Science, Springer-Verlag (August 2000).

[7] Colin Boyd and Dong Gook Park: Public Key Protocols for Wireless Communications ; Proceedings of ICISC'98, Korea Institute of Information Security and Cryptology, pp.47-57.

[8] D. Davis: Kerberos Plus RSA for World Wide Web Security; Proc. 1st USENIX Workshop on Electronic Commerce, NYC, July 1995.

[9] J. Feigenbaum: Locally Random Reductions in Interactive Complexity Theory; in Advances in Computational Complexity Theory, DIMACS Series on Discrete Mathematics and Theoretical Computer Science, volume 13, American Mathematical Society, Providence, 1993, pp. 73--98.

[10] M. Franklin, D. Boneh: Efficient generation of shared RSA keys; Advances in Cryptology -- Crypto '97 Proceedings

[11] M. Franklin, D. Coppersmith, J.Patarin, M. Reiter: Low exponent RSA with related messages; Advances in Cryptology -- Eurocrypt '96 Proceedings. Earlier version in IBM Research Report RC 20318, December 27, 1995

[12] osario Gennaro, Hugo Krawczyk, Tal Rabin: RSA-based Undeniable Signatures; CRYPTO'97

[13] O. Goldreich, S. Goldwasser, S. Halevi: Public-Key Cryptosystems from Lattice Reduction Problems; To appear in CRYPTO '97. Available as ECCC Report TR96-056.

[14] M. Bellare, A. Boldyreva, S. Micali: Public-key Encryption in a Multi-User Setting: Security Proofs and Improvements ; Extended abstract in Advances in Cryptology - Eurocrypt 2000 Proceedings, Lecture Notes in Computer Science Vol. ??, B. Preneel ed, Springer-Verlag, 2000.

[15] J. Seberry, X. M. Zhang, Y. Zheng: Structures of cryptographic functions with strong avalanche characteristics; Advances in Cryptology -- AsiaCrypt'94, Lecture Notes in Computer Science, Vol.917, pp.119-132, Springer-Verlag, 1995.

[16] V. Shoup: On the deterministic complexity of factoring polynomials over finite fields; Information Processing Letters 33:261-267, 1990.

A New Multisignature Scheme Based on Discrete Logarithm Problem and its Distributed Computation

Lu Langru[1] Zeng Junjie[1] Kuang Youhua[2] Cheng Shengli[3]
[1.] Information Security Lab., Information Engineering Uni.
ZhengZhou. PRC. 450002
[2.] Math. Section, LuoYang Foreign Language Institute.
[3.] The Dept.of Computer, HuaZhong Uni. of Sci. and Tec.
infsecl@public2.zz.ha.cn

ABSTRACT

We show an attack for purpose on most multisignature schemes of which extend from signature Schemes based on DLP such as Meta-El Gamal multisignature scheme [1] and Schnorr multisignature scheme [2]. The attackers can deny that they had taken part in process of signing some message with others. A modification is made for these schemes' key generations, which can efficiently void this attack. We also design a method to distributed computation of this new multisignature scheme in this paper.

Keywords: Multisignature, Deny, Discrete Logarithm Problem, Distributed Computation

1. INTRODUCTION

Electronic commerce is becoming a motivation for increase of economic in 21'st Century, while security problem is a bottleneck of reliable environment for it. It was estimated that eighty per cent of Internet trade information required content integrity, identity authentication and deny-resistant protection, which digital signature, one key technology of e-commerce, exactly provides these function.

Digital signature is playing an important pole in our information society. In many circumstances there need not only individual signature, but also signatures which some people sign a message with cooperation, the so-called multisignature. For example, committee of presidents holds a meeting to discuss the plan of company's future development in Internet. They can use multisignature technology to sign the resolution. Some commerce department is qualified to carry on some business by gaining multisignatures.

A former and simple solution to multisignatures is that everyone signs the same message with the same signature scheme and then they collect all their signatures respectly as multisignature for the message. No cooperation exists in the signing process. Length of the multisignature is unsatisfactory in that almost increases linearly with the increase of the number of signers. Latterly multisignature schemes based on DLP are proposed such as scheme based on Meta-ElGamal scheme [1], scheme based on Schnorr scheme [2]. Moreover, the authors proved their security equivalent to those of signature schemes with single signer, assumed all participants obeyed protocols (see [2]). But it is too restricted and idealized. We find if some signers copper together during key generations they can deny their relation to multisignatures on any message. So we modify keys generating of these multisignature schemes and void the attack.

In section 2 we review two basic multisignature schemes above; Section 3 describes our attack in detail; Section 4 shows our modification; Section 5 designs a method to distributed computation of Schnorr multisignature scheme. Finally, a short conclusion is given in section 6.

2. BASIC MULTISIGNATURE SCHEMES BASED ON DLP

Two basic schemes are given here, which mainly shows the generation of our attack.

2.1 Meta-ElGamal_based multisignature scheme

(1) Key generation
A trusted center publishes a large prime p and a primitive α in Z_p^* as system public parameters. Signer P selects randomly an integer $x \in Z_{p-1}^*$, computes $y = \alpha^x \bmod p$. Her private key and public key are x and y respectively. All these value are the same for any message to sign. Assume signer P_i has private key x_i and public key y_i (i=1,2,...,n).

(2) Multisignature generation
To sign a message m, the n signers execute the following procedure:

Step 1) Signer P_i randomly selects an integer $k_i \in Z_{p-1}^*$, and computes $r_i = \alpha^{k_i} \bmod p$ (for i=1,2,...n), then broadcasts it to other signers. So each signer calculates:

$$r = \prod_{i=1}^{n} r_i \bmod p .$$

Step 2) Signer P_i calculates her signature parameter S_i as follows: $S_i = x_i(m+r) - k_i \bmod p$ for i=1,2,...n, then sends it to a employee Clerk who knows m and r.

Step 3) The main task of Clerk is to verify each single signature S_i by checking the following modular equation:

$y_i^{m+r_i} = r_i \cdot \alpha^{S_i} \bmod p$.

If the equation doesn't hold for some i ($1 \le i \le n$), Clerk will halt the procedure and inform all signers.

Step 4) If the equations hold above for all i, Clerk calculates $S = \sum_{i=1}^{n} S_i \bmod p - 1$.

3-dimension vector (m, r, S) is multisignature for message

m from signers $\{P_1,P_2,\ldots,P_n\}$.

(3) Multisignature verification
A verifier verifies the message with the multisignature (m,r,S) from signers $\{P_1,P_2,\ldots,P_n\}$ by checking the following equation: $y^{m+r} = r \cdot \alpha^S \bmod p$

where $y = \prod_{i=1}^{n} y_i \bmod p$.

If the above equation holds, the verifier accepts the multisignature as valid; Otherwise, refuses it.

2.2 Schnorr_based multisignature scheme

The article [2] describes two schemes: simultaneous multisignature scheme and sequential one. Since they are the same from a view of technology, we only review the latter.

(1) Key generation
A center selects two large primes p and q, which satisfy $q|p-1$, $q \geq 2^{140}, p \geq 2^{512}$, and an element α in Z_p with order q, namely $\alpha^q = 1 \bmod p, \alpha \neq 1$. It also selects a one-way hash function $h: Z_p \times Z \to \{0,1,\cdots,2^T-1\}$, where T is a security parameter. Then it publishes p and q, α and h as lasting system parameters.
Signer P_i randomly selects an integer $x_i \in Z_q^*$, computes $y_i = -\alpha^{x_i} \bmod p$. Her private key and public key are x_i and y_i respectively (i=1,2,...,n).

(2) Multisignature generation
To sign a message m, the n signers execute the following procedure:
Step 1) Repeat while i=1,2,...,n
Signer P_i receives e_{i-1}, where $e_0 = 1$ holds, then generates a random integer $k_i \in Z_q^*$, calculates
$e_i = e_{i-1} \cdot \alpha^{k_i} \bmod p$, and sends e_i to the next signer P_{i+1} (where denotes $P_{n+1}=P_1$).
Step 2) Repeat while i=1,2,...,n-1
Signer P_i receives (m,e_n,S_{i-1}) from P_{i-1}, where $s_0 = 0$ holds, then calculates $e = h(e_n,m)$ and S_i as follows:
$S_i = S_{i-1} + (k_i + x_i \cdot e) \bmod q$, and sends (m,e_n,S_i) to the next signer P_{i+1}.
Step 3) Signer P_n receives (m,e_n,S_{n-1}) from P_{n-1}, calculates $e = h(e_n,m)$ and
$S_n = S_{n-1} + (k_n + x_n \cdot e) \bmod q$
S_n as follows:
Denote $S = S_n$. 3-dimension vector (m,e,S) is multisignature for message m from signers $\{P_1, P2\ldots Pn\}$.

(3) Multisignature verification
A verifier verifies the message with the multisignature (m,e,S) from signers $\{P_1, P2\ldots Pn\}$ as follows:
1) The verifier calculates $\bar{e} = \alpha^S \cdot \left(\prod_{i=1}^{n} y_i\right) \bmod p$;
2) The verifier calculates $h(\bar{e},m)$, then checks whether the equation $e = h(\bar{e},m)$ holds.
If the equation holds, he accepts the multisignature; otherwise refuse it.

3. CRYPTANALYSIS OF BASIC MULTISIGNATURE SCHEME

We first give a variant of attack proposed in [3]: Assumed attacker party consists of signer P_i (i=1,2,....,t), they sign the message m together with signer P_j (j=t+1,t+2,...,n) in order to forge multisignatures, according to two basic multisignature schemes. They can execute the following procedure:

1) Attacker party generates their private keys x_i (i=1,2...t) which satisfy
$\sum_{i=1}^{t} x_i \equiv 0 \bmod p-1$
or $\sum_{i=1}^{t} x_i \equiv 0 \bmod q$ (*)
according to the two basic schemes.

2) Attacker party disobeys the schemes and generates their $k_i \in Z_{p-1}^*$ (i=1,2,...t) with cooperation, which satisfy
$\sum_{i=1}^{t} k_i \equiv 0 \bmod p-1$
or $\sum_{i=1}^{t} k_i \equiv 0 \bmod q$

3) Attacker party with signer P_j (j=t+1,t+2,...,n) signs the message m together and sends multisignature (m,r,S) or (m,e,S_n) to verifier(or receive).
It is obvious that (m,r,S) or (m,e,S_n) can accept as the multisignature from both signers $\{P_1,P_2,\ldots,P_n\}$ and signers $\{P_{t+1},P_{t+2},\ldots,P_n\}$.
Assumed unconditional trust among attacker party, they can easily decide their private keys x_i and k_i together.
Assumed no unconditional trust among attacker party, [3] describes a method to calculate k_i for P_i (i=1,2,...,t) without the knowledge of it, which obviously works to calculate x_i for P_i.
If no technology modifications are made for the basic schemes, attacker party can get multisignatures from signers $\{P_{t+1},P_{t+2},\ldots,P_n\}$. [3] shows a likely method to the attack which each signer checks all products of subset of $\{r_1,r_2,\cdots,r_n\}$ or $\{e_1,e_2,\cdots,e_n\}$ don't equal to one. But the number of subsets is $\sum_{i=1}^{n-1}\binom{i}{n} = 2^n - 2$, which increases exponentially with the increase of the number of signers. A better solution is that messages concatenating the number of signers or identities of signers will be signed. [3] also gives an idea that a hash function h taken signer's unique token ID as parameter will be used, namely $hash(m, ID_1, ID_2, \cdots, ID_n)$.
Notice the number of signers, identities or ID are public, if purpose of attacker party adjusts to deny multisignatures when messages become disadvantageous to it latter, it can execute a variant of attack above which is more hideaway and threatening. We describes it as follows:

1) Attacker party includes signing-attacker party $\{P_1,P_2,\ldots,P_t\}$ and a coppering-attacker P with private

keys $\{x_1, x_2, \cdots, x_t\}$ $\{k_1, k_2, \cdots, k_t\}$ and x , k respectively which satisfy

$\sum_{i=1}^{t} x_i + (p-1-x) \equiv 0 \bmod p-1$,

$\sum_{i=1}^{t} k_i + (p-1-k) \equiv 0 \bmod p-1$,or

$\sum_{i=1}^{t} x_i + (q-x) \equiv 0 \bmod q$, $\sum_{i=1}^{t} k_i + (q-k) \equiv 0 \bmod q$ (**)

2) Signing-attacker party signs the message m with other signers P_j (j=t+1,t+2,…,n), according to the basic schemes.

3) Signing-attacker party can get multisignature for message m from {P_1, P_2,…,P_n} which can be forged easily by {P, P_{t+1}, P_{t+2}, …, P_n}. So they can deny it in court latter and have a policy that P admits to signing with P_j (j= t+1, t+2,…,n) at the price of himself when necessary.

The methods above don't work to resist the variant of attack above.

4. MODIFIED MULTISIGNATURE SCHEME AND THEIR SECURITY ANALYSIS

We consider the weakness of basic multisignature schemes as key generations by Signers alone. A secure form of key generation should combine with cooperation of center and signer. Moreover, private keys are still random. As an example, we describe our modification of basic Schnorr-based multisignature scheme in its key generation that can void the attack above.

A center selects two large primes p and q, which satisfy $q|p-1$, $q \geq 2^{140}$, $p \geq 2^{512}$, and an element α in Z_p with order q, namely $\alpha^q \equiv 1 \bmod p$, $\alpha \neq 1$. It also selects a one-way hash function $h: Z_p \times Z_p \rightarrow \{0,1,\cdots,2^T-1\}$, Where T is a security parameter. Then it publishes p and q, α and h as lasting system parameters.

(1)Assumed an absolutely trusted center, during key generation signer P_i (i = 1,2,⋯, n) selects randomly an integer $x_{p_i} \in Z_q^*$, computes $y_{p_i} = -\alpha^{x_{p_i}} \bmod p$, and sends y_{p_i} to center; Center also selects randomly $l_i \in Z_q^*$, computes $y_i = y_{p_i} \cdot \alpha^{l_i} \bmod p$, and sends l_i and y_i to P_i; P_i calculates $x_i = x_{p_i} + l_i \bmod q$ as her private key and y_i as her public key; Finally, center publishes y_{p_i}, l_i and y_i; Other parts of basic Schnorr-based multisignature scheme wouldn't be altered.

(2)Assumed no absolutely trusted center, center publishes another one-way function $f: Z_p \times Z_p \rightarrow \{0,1,\cdots,2^N-1\}$, where N is a security parameter. During key generation signer P_i (i = 1,2,⋯, n) randomly selects an integer $x_{p_i} \in Z_q^*$, computes $y_{p_i} = -\alpha^{x_{p_i}} \bmod p$, and sends y_{p_i} to center; Center also selects randomly an integer $l_i \in Z_q^*$, computes $y_i = y_{p_i} \cdot \alpha^{f(y_{p_i}, l_i)} \bmod p$, and sends l_i and y_i to P; P calculates $x_i = x_{p_i} + f(y_{p_i}, l_i) \bmod q$ as her private key and y_i as her public key; Finally, center publishes y_{p_i}, l_i and y_i which can be verified by other signers; Other parts of basic Schnorr-based multisignature scheme wouldn't be altered.

So the attacker party {P_1, P_2,…,P_t} or {P,P_1,P_2,…,P_t} can't gain their private keys satisfying (*) or (*) even with the aid of center. Because of no alteration in other parts of basic Schnorr-based multisignature scheme except key generation, security of new scheme equals to security of basic Schnorr-based multisignature scheme.

Obviously private keys of modification (1) are random; Since $f(y_{p_i}, l_i)$ is interrelated to x_{p_i}, it can't assume randomness of private keys of modification (2)(i.e. uniform distribution in key space); f must be a one-way function to protect attacker party {P_1, P_2,…,P_t} or {P,P_1,P_2,…,P_t} from calculating their private keys satisfying (*) or (*) because it and center have no enough computers to calculate inverse; So selection of function f is very important.

Basic Meta-ElGamal-based multisignature scheme and simultaneous Schnorr-based multisignature scheme, also other multisignature schemes based on DLP have similar modifications.

5. DISTRIBUTED COMPUTATION OF NEW SCHEME

Transmitting time of signing sequential multisignature is proportional with the number of signers. Community and calculation of each signer is almost same, which called loading balance. To improve signing speed, we design a binary tree structure of distributed computation of multisignature based on Schnorr scheme as an example. Assumed the number of signers $n = 2^k - 1$, we found k layers binary tree structure of signers(see Figure 1), where P is signing sponsor. We also assume each signer knows other signers and message m; otherwise , P executes step1 and informs other signers of m and the binary tree structure.

Denote s as a string, $|s|$ as length of s (when $|s|$=0, s is an empty string ε), P_s as P_{ε} ($|s|$=I)as a signer infecting a node in i'th layer.

Step1 Each signer executes some algorithm (for example, fixed P as root node , considered IP address as an integrity, each signer orders all IP) to found the structure .They use it until signing process is over and only needs to know their father nodes and sun nodes.

Step2 Calculating e value

Step2.1 Signer P_s in the i'th layer nodes ($|s|$=i) selects randomly integer k_s and computes $e_s = \alpha^{k_s} \bmod p$;

Step2.2 Signers P_{s0} and P_{s1} ($|s|$=i, i=k-2,k-3,…,0) transmit e_{s0} and e_{s1} totheir father node P_s .P_s calculates $e_s = e_s \cdot e_{s0} \cdot e_{s1} \bmod p$. Finally P gains e_{ε}.

Step2.3 Signers transmit m and e_ε from the 0'th layer to the k'th layer according to opposite order in Step2.2. When P_s receives m and e_ε ,he calculates $e = h(e_\varepsilon, m)$ and $S_s = k_s + x_s \cdot e \bmod q$.

Step3 Signers P_{s0} and P_{s1} ($|S|$=i, i=k-2,k-3,...,0) transmit S_{s0} and S_{s1} to P_s. Then P_s calculates $S_s = S_s + S_{s0} + S_{s1} \bmod q$. Finally P gains the integer signature on m $S = S_\varepsilon$ and transfers it to verifier.

It's clears that signing time of multisignature equals to that of maximal transmitting delay of some path starting from P. We assume all transmitting delay between father nodes and their sun nodes. Comparing with sequential multisignature , each signer calculates more one time in Step2.2 and Step3, transmits more one time in Step2.3; our algorithm speed is nearly its $2(n-1)/(3\log n)$ times. Comparing with concentrated transmit, that is each signer transfers his all information to sponsor P, our algorithm won't lead to bottleneck because of its loading balance.

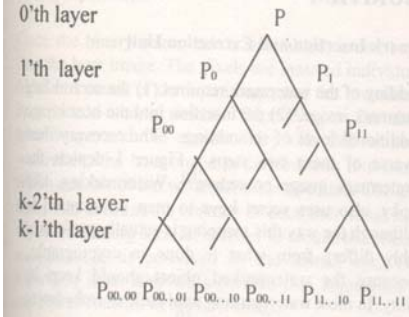


Fig.1 Binary tree structure

6. CONCLUSIONS

Most multisignature schemes present based on DLP extend from signature schemes with single signer, their security is equivalent to former signature schemes with single signer under some hypothesis. For some properties of finite fields key generations of these multisignture schemes have changed. Our work is worth referring to research of multisignature schemes and its applications. Problem of construction of function f is to get into. We also design a method to distributed computation of Schnorr multisignature scheme.

REFERENCES

[1] L . Harn ,"New digital signature scheme based on discrete logarithm", Electronics Letters , Vol.30.No.5. (1994). pp.396 – 398.

[2] Ji Jiahui ,Zhao Renjie,"Digital Multisignature Schemes Based on the Dchnorr Scheme", 密码学进展— CHINACRYPT' 96, pp. 170-176

[3] P.Horster,M.Michels,and H.Peterson, Meta-multisignature schemes based on the discrete logarithm problem, IFIP/Sec'95, pp128-142.

Research on an Adaptive Digital Image Watermarking Technique

Jie Yang
School of Electronics and Information Technology, Wuhan University of Technology
Wuhan 430063, China Email: jieyangz@public.wh.hb.cn

Moon Ho Lee
Institute of Information & Communication Chonbuk National University
Chonju, 561-756, Korea

ABSTRACT

Digital watermarking techniques have been proposed as an effective solution to the protection of copyright of multimedia data. Nevertheless, the success of watermarking in copyright protection applications depends on the technical possibility of satisfying the requirements imposed by practical applications. In this paper, an adaptive method for digital image watermarking is presented. This technique produces a watermarked image that closely retains the quality of the original host image while concurrently being robust to various image processing applications such as lowpass and median filtering, image scaling, lossy JPEG compression, cropping and rotation.

KeyWords: Digital Watermarking, Image Processing, Copyright Protection

1. INTRODUCTION

Many of the problems that we need to solve in enforcing copyright law for digital content are similar to problems in secure communications that have been solved using cryptography. To be effective in the protection of the ownership of intellectual property, the invisibly watermarked document should satisfy several criteria: the watermark must be difficult or impossible to remove, at least without visibly degrading the original image, the watermark must survive image modifications that are common to typical image-processing applications (e.g., scaling, color re-quantization, dithering, cropping, and image compression), an invisible watermark should be imperceptible so as not to affect the experience of viewing the image, and for some invisible watermarking applications, watermarks should be readily detectable by the proper authorities, even if imperceptible to the average observer.

Such decodability without requiring the original, un-watermarked image would be necessary for efficient recovery of property and subsequent prosecution [1].

This paper presents an adaptive method for digital image watermarking for copyright protections that is a spatial domain technique. The experimental results show that the proposed method is robust against the attack of lowpass and median filtering, lossy JPEG compression and rotation. Additional features of this technique include the easy determination of the existence of the watermark by human visual inspection and a double encryption scheme.

2. DESCRIPTION OF ADAPTIVE DIGITAL IMAGE WATERMARKING TECHNIQUE ALGORITHM

a) Watermark Insertion and Extraction Unit

The embedding of the watermark requires (1) the scrambling of the watermark image, (2) the insertion into the host image, with an additional level of scrambling. The recovery then is the reverse of these two steps. Figure I depicts the general watermark image procedure[2]. Watermarking, like cryptography, also uses secret keys to map information to owners, although the way this mapping is actually performed considerably differs from what is done in cryptography, mainly because the watermarked object should keep its intelligibility. In most watermarking applications embedment of additional information is necessary [3]

b) Watermarking Algorithm Steps

In the watermark forming process, the pixels of the watermark are pseudo-randomly permuted to form a new watermark image for the scrambling of the watermark,. The pseudo-random permutation is can be done using a linear feedback shift register. By setting the state of the shift register, a pseudo-random sequence can be generated that is then recoverable by resetting the shift register to its original state. The shift register can be applied in two fashions. The first option is to use the shift register to generate a random sequence of new row and column indices for the two dimensional watermark. This option requires repeatedapplications of the shift register for both the row and column indices. This is due to the fact that the row and column indices must fall in the range given by the size of the watermark image. Thus, an entirely new set of row indices must be generated for a single column index. The second option is more direct and easier to implement. First, a raster scan of the watermark image is performed to generate a single row vector from the watermark. The elements of this row vector can then be pseudo-randomly permuted into a new row vector via a single execution cycle of the linear shift register.

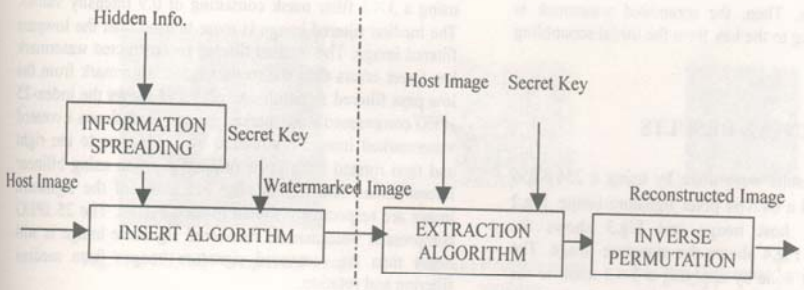


Fig.1: Watermark Insertion and Extraction Unit

The shift register must only perform one permutation of the indices of the raster scan vector. A new raster scan vector is then generated by assigning the elements from the old raster scan vector to the positions of the new vector, as given by the newly generated indices. The scrambled watermark is then constructed by performing the inverse raster scan process on this vector. This second method is implemented in this experiment.

Once the binary watermark is scrambled, it can be inserted into the host image. The pixels are inserted individually into blocks of pixels of the host image, and the insertion is done in a pseudo-random fashion. Depending on the size of the host image and the number of pixels that make up the watermark, the host image is divided into n×n blocks into which one bit of the watermark is embedded. The algorithm takes into account the contrast of each individual n×n block when embedding each bit in order to reduce the effects of the modifications as perceived by the human eye. Thus, the algorithm is an adaptive algorithm that adjusts for varying regions of contrast in the host image.

The bits from the scrambled watermark are selected for embedding, again in raster scan order, from the watermark image. A single bit is encoded into one n×n block of the host image, and the nxn block is selected from a randomly permuted set of indices that index the nxn blocks throughout the host image. Once a bit from the scrambled watermark, b_w, and a 4×4 block, B, into which it will be embedded have been selected, the bit is inserted in the following Step:

Sort the pixels in block B in an ascending order based on their intensity values.

Compute the average, g_{mean}, minimum, g_{min}, and maximum, g_{max}, of the intensities of the pixels in B.

Classify each pixel into one of two categories, based on whether its intensity value is above or below the mean of the block, i.e., the ijth pixel, bij is classified depending on its intensity, g, as

$$b_{ij} \in Z_H \quad \text{if } g > g_{mean} \tag{1}$$

$$b_{ij} \in Z_L \quad \text{if } g \le g_{mean} \tag{2}$$

Where Z_H and Z_L are the high and low intensity classes, respectively.

(4) Compute the means, m_L and m_H, for the two classes, Z_L and Z_H.

(5)Define the contrast value of block B as

$$C_B = \max(C_{min}, a(g_{max}-g_{min})) \tag{3}$$

Where a is a constant and C_{min} is a constant which defines the minimal value a pixel's intensity can be modified.

(6) Given the value of bw is 0 or 1, modify the pixels in B according to:

if $b_w = 1$,

$$g_{new} = g_{max} \quad \text{if } g > m_H \tag{4}$$

$$g_{new} = g_{mean} \quad \text{if } m_L \le g < g_{mean} \tag{5}$$

$$g_{new} = g + \delta \quad \text{otherwise} \tag{6}$$

if $b_w = 0$,

$$g_{new} = g_{min} \quad \text{if } g < m_L \tag{7}$$

$$g_{new} = g_{mean} \quad \text{if } g_{mean} \le g < m_H \tag{8}$$

$$g_{new} = g - \delta \quad \text{otherwise} \tag{9}$$

where g_{new} is the new intensity value for the pixel which had original intensity value g and δ is a random value between 0 and C_B.

(7) The modified block of pixels, B_{new}, is then positioned in the watermark image in the same location as the block, B, of pixels from the original host image. These steps describe the procedure by which the watermarked image is generated from a host and a watermark. Thus, the pixels are modified in a manner that is adaptive to the contrast value of the regional block of pixels. The result is that, if a 1 is embedded into a block, the average intensity value for that embedded block will be greater than the average intensity for the same block of the original host image. If a 0 is embedded, then the average intensity of the embedded block will be lower than that of the original host. By using the offset δ, those pixels which it modifies will have a small random noise component, however with a nonzero overall mean value. The random nature of this tuning helps to prevent a visible blocking effect while still contributing to the shift of the overall mean of the block of pixels. This also contributes to the robustness of the algorithm to some of the image filtering processes while also reducing the blocking. The filtering that might be performed on the watermarked image may reduce the variance of the noise, however, given its nonzero mean, the average may still be preserved at a higher level for a given block.

c) Extracting the Embedded Watermark

The extraction algorithm is straightforward and requires the original host image. The extractor need only compute the sum of the intensity values for the blocks of the host and watermarked image. A bit is decoded by making the comparison of the two resultant values:

$$\text{if } S_w > S_o, \text{ then } b_w = 1 \tag{10}$$

$$\text{if } S_w \le S_o, \text{ then } b_w = 0 \tag{11}$$

where S_w and S_o are the sums for the blocks of the watermarked and original images, respectively. The decoded bits are then entered into the inverse permuted order as the nxn blocks were selected by using the key from the scrambled insertion procedure. This produces the recovered

scrambled watermark. Then, the scrambled watermark is descrambled according to the key from the initial scrambling operation.

3. EXPERIMENTAL RESULTS

The experimental results were done by using a 256×256 pixel host image and a 64×64 pixel signature image. Fig.2 shows the original host image and Fig.3 shows the watermarked image Fig.4 shows the signature image. The median filtering was done by applying a 3×3 mask to the watermarked image. The median filtered image is shown in figure 5. Fig.6 shows a lowpass filtered watermarked image using a 3×3 filter mask consisting of 0.9 intensity values. The median filtered image is more blurred than the lowpass filtered image. The median filtered reconstructed watermark has fewer errors than the reconstructed watermark from the low pass filtered signature image. Fig.7 shows the index-25 JPEG compressed watermarked image. Fig.8 shows a rotated watermarked image. It's rotated by 17 degrees to the right and then rotated back to its original position using bilinear interpolation. The results after extraction of the signature image are respectively shown in each Figure. The 25 JPEG compressed watermarked recovered signature image is still better than the recovered signature images from median filtering and rotation.

Fig.2 Original image

Fig.3 Watermarked image

Fig.4 Original signature image

Fig.5 (a) Median filtered watermarked image using a 3×3 mask

Fig.5 (b) Reconstructed signature

Fig.5(c) Difference between the original and reconstructed signature

Fig.6 (a) Low pass filtered watermarked image using a 3×3 mask

Fig.6 (b) Reconstructed signature

Fig.6(c) Difference between the original and reconstructed signature

Fig.7 (a) JPEG 25 compressed image

Fig.7(b) Reconstructed signature

Fig.7(c) Difference between the original and reconstructed signature

Fig.8(a) Watermarked image rotated 17 degree

Fig.8(b)Reconstructed signature

Fig.8(c) Difference between the original and reconstructed signature

4. CONCLUSION

The growth of networked multimedia systems has magnified the need for image copyright protection. One approach used to address this problem is to add an invisible structure to an imge that can be used to seal or mark it. These structures are known as digital watermarks. In this paper we implement the algorithm, which is robust to low pass filtering, median filtering, rotation, and lossy JPEG compression. This means that an embedded signature image is still recoverable and recognizable of the owner after the watermarked image has been tampered with or modified by common image processing techniques.

REFERENCES

[1] Koch E.Zhao J. Embedding robust labels into images for copyright protection. Technical Report Fraunhofer Institute for Computer Graphics, Darmstadt, Germany, 1994.

[2] Gunnar Saanum Gulstad, Kristoffer Bruvold, An Adaptive Digital Image Watermarking Technique For Copyright Protection, University of California, Santa Barbara, 1999

[3] Xiamu Niu, Zheming Lu and Shenghe Sun, Digital Watermarking of Still Image with Gray-Level Digital Watermarking, IEEE Trans. On Consumer Electronics, Vol., No.1, Feb. 2000

Parallel Acceleration of AES Algorithm Using FPGA

Lu Langru[1] Kuang Youhua[2] Yang Qianghao[1] Cheng Shengli[3]
[1.] Information Security Lab., Information Engineering Uni.
ZhengZhou, PRC.450002
[2.] Math. Section, LuoYang Foreign Language Institute.
[3.] The Dept.of Computer, Huazhong Uni. of Sci. and Tec.
infsecl@public2.zz.ha.cn

ABSTRACT

The paper discusses how to implement and parallel accelerate AES using FPGA with SPARTANII structure of XILINX . Gives some parallel ways to increase the speed of AES, and discusses some questions they bring. At the end, estimates the speed target that AES algorithm could reach.

Keywords: AES, FPGA, Parallel Accelerate, Pipeline

1. BACKGROUND AND BASIC PARAMETER

The National Institute of Standards and Technology (NIST) has been working with industry and the cryptographic community to develop an Advanced Encryption Standard (AES). The overall goal is to develop a Federal Information Processing Standard (FIPS) that specifies an encryption algorithm(s) capable of protecting sensitive government information well into the 21 century. On January 2, 1997, NIST announced the initiation of the AES development effort and made a formal call for algorithms on September 12, 1997. On October 2, 2000, NIST announced that it has selected Rijndael to propose for the AES. It is anticipated that the standard will be completed by the summer of 2001,and put into use in the summer.

Rijndael is a block cipher, designed by Joan Daemen and Vincent Rijmen as a candidate algorithm for the AES .The cipher has a variable block length and key length. We currently specified how to use keys with a length of 128, 192, or 256 bits to encrypt blocks with al length of 128, 192 or 256 bits (all nine combinations of key length and block length are possible). Both block length and key length can be extended very easily to multiples of 32 bits. Rijndael can be implemented very efficiently on a wide range of processors and in hardware. Rijndael can be implemented very efficiently on a wide range of processors and in hardware. The design of Rijndael was strongly influenced by the design of the block cipher square [7].

We have been researching the AES algorithm in the late 2000.The AES algorithm has been implemented in three respect: software (coded in c language), firmware (16-bit parallel fixed DSP), hardware (FPGA). The achievement is list as following:

- The estimated result of parallel compute performance in theory

The estimated result of in the 8-bit mode:

KEY LENGTH	CIPHER SPEED	DECIPHER SPEED
128	9.5 Mb/s	2.6 Mb/s
192	9.0 Mb/s	2.4 Mb/s
256	8.5 Mb/s	1.9 Mb/s

The estimated result of in the 32-bit parallel mode (with 4 tables):

KEY LENGTH	CIPHER/DECIPHER SPEED
128	112.9 Mb/s
192	106.4 Mb/s
256	100.6 Mb/s

- Test result of software implement

Test under single processor and single user, the configuration of PC is:

- CPU: Celeron 266 Mhz
- MEMORY: 32M
- OPERATING SYSTEM: RedHat 6.1 Linux (Kernel 2.2.12-20)

Test result under 8-bit mode

Key Length	CIPHER		Decipher		Key Extension
	CLOCK	Mb/s	CLOCK	Mb/s	CLOCK
128	12 378	2.6	37 251	0.8	9 056
192	14 788	2.2	45 229	0.7	11 433
256	17 178	1.9	52 895	0.6	15 502

Test result under 32-bit mode

Num-ber of Table	Key Len-gth	CIPHER/ DECIPHER		Key Exten-sion	Athwart Key Extension
		CLOCK	Mb/s	Clock	Clock
1	128	695	46.7	595	3 556
	192	825	39.3	601	4 209
	256	958	33.8	751	4 985
4	128	614	52.9	591	3 553
	192	731	44.4	601	4 205
	256	841	38.6	747	4 981

- Test result of firmware

Test environment:
TI (Texas Instruments) corporation:
TMS320C6201B-200; Outside oscillator:40MHz;
phase-locked loop: ×5;
Parallel degree: 8

Test result under 8-bit:

Number of Table	Key Len -gth	Cipher/ Decipher	Key Exten -sion	Athwart Key Extensio n
		Clock Mb/s	Clock	Clock
1	128	281 74	614	3012

2. STRONGPOINT OF PGA IMPLEMENTATION OF AES AND PARALLEL ACCELERATION

In the propose of AES, the problem of parallel procession has been considered. But it is only limited in the common processor mode. In the view of parallel acceleration, recently any common processor mode computes a block concurrent (either 128-bit or 256-bit) . In the aspect of the processor's process, at any moment, the processor only process a part of data in one step of the algorithm. As every round need the mediate result of former round, encipher/decipher in the whole process, a lot of resource in the idle state. The speed has been affected.

The problem can been resolved by hardware implementing with FPGA. Because AES is different from the public-key algorithm, such as RSA. There is little complex computation (such as multiple, mod, etc).In the parallel compute process, the request of the structure of PMPU(Parallel Micro Calculate Unit) is relatively simple, and it only has logic compute and search compute. The function of every feasible processor unit is too powerful, while in the FPGA multi-processor based on AES can been designed. and every bit of one block compute at the same time, it will increase the resource utilization.

3. THE IMPLEMENT OF AES IN FPGA

Considering parallel acceleration need much hardware resource (such ass routing resource and register), we select FPGA with SPARTANII structure of XILINX corporation as carrier of AES. For be easy to transplant to other carrier, we deal AES with VHDL. The result is validated on DEMO board designed by us.

The implement consists of two components: encrypt/decrypt and key expansion. Only encrypt be discussed in the encrypt/decrypt part. The next discuss is based on 128bits key length and 192bits data block.

1). Encrypt

The cipher consists of 12 rounds, the operation is some same in every rounds: taking inverse, apply an affine (over GF(2)) transformation, ShiftRow transformation, MixColumn transformation, Key EXOR.

- Taking inverse, apply an affine(over GF(2)) transformation : the two step is one-one mapping over $GF(2^8)$, can calculate a table with 256 bits. The structure of SPARTANII in Xilinx is CLB search table structure, which is easy to implement by search table operate, and it can complete in 2-3 CLB. There is 8bits data can calculated in one table, Construct 24 same tables (over GF(2)) (aim at 192bits block), ensure that all data in one block is complete in same time(CLB's delay time is invariableness).
- ShiftRow transformation: the operation of ShiftRow transformation is only cyclically shifted data by byte, Only adjust at routing, it occupy no source of FPGA basically, and the delay time can ignore (in the essential time we can intervene the routing inside of FPGA by manual).
- MixColumn transformation: It can considered as polynomials, given by

  $B_0 = 02 (A_0 \oplus A_1) \oplus (A_1 \oplus A_2 \oplus A_3)$
  $B_1 = 02 (A_1 \oplus A_2) \oplus (A_2 \oplus A_3 \oplus A_0)$
  $B_2 = 02 (A_2 \oplus A_3) \oplus (A_3 \oplus A_0 \oplus A_1)$
  $B_3 = 02 (A_3 \oplus A_0) \oplus (A_4 \oplus A_1 \oplus A_2)$

  Accord with the formula we can organize the MixColumn to logic operation. There is same MixColumn Operation of every column so that this step can complement at same time.
- Key EXOR: this step is EXOR by bit, before process the step operation, the expanded key is be ready.

2). Key expansion

The operation of key expansion consists of: taking inverse, apply an affine(over GF(2)) transformation, shift of the byte, calculate round constants, calculate key.

- Taking inverse, apply an affine(over GF(2)) transformation, shift of the byte: the three steps forward is same as the operation of cipher.
- Calculate round constants: round constants can make table after calculate by the max round length.

Calculate key: EXOR operate.

4. PARALLEL ACCELERATION BASED ON AES CHARACTERISTIC

Parallel acceleration is involved when AES is designed. We take advantages of it in implement using FPGA while it can bring about some difficulty in fact. Following illustrate how to utilize it and steps to solve the relative problems in detail.

- Paralleling of one round and matching of task load

  We can use twenty four GF(2) searching tables and six combined calculators in a single encryption round to guarantee all the bit-calculations in one block can be done almost simultaneously.(In FPGA according to searching tables, delay is concerned only with the number of tables need to search and inside routing delay.)

  Xor round key is the final in each round encryption .Key extension can go with encryption simultaneously. So there comes one problem—matching of task load. It is an ideal result that encryption (excluding xor a round key) needs more time than key-extension in a same round. We can learn from the discussion of the 3rd part that searching tables and various combinations are

included in encryption. Key extension only need searching tables if there is a round constant. And following operation is XOR. We can take searching-result, round constant, key and encryption result in last round together to XOR (It is faster than multiple- XOR)

The creation of a round key saves at least one CLB contrast with encryption-delay. We can get balance of task load in one round. Only asynchronized logic calculations are implemented to speed up in each round. Round constant is created when FPGA is powered on and saved in block RAM. It takes up 144 bits total. Block RAM will be discussed in the next part.

- Once extend all keys at first

AES has several length of key, the process of key expansion is same. Usually key will be bring into FPGA first, then data will be in.. After key is in, could be once extended with the max length and saved in chip. When need, take some of them. The speed of once extend all keys at first is faster than synchronous extension. They need no more 1536 bits of RAM.

- Competition and risk

Nearly th same operations are needed in each round of AES. They are implemented asynchronous so that bring competition and risk. In every round, main operations are search in some table, The results are got nearly at the same time. They do not bring competition and risk. Competition and risk are created by MixColumn transformation and Key EXOR. In order to not take competition and risk into the next round, the results of Key EXOR must be saved in D register. Using the global clock, the result of this round could be bring to next round at the same time.

But it gives a new question: how to decide the delay of global clock? If the length of delay is longer, the speed of AES will be decrease. If shorter, the results will not bring to next round correctly

Delay of D register = Delay of a round operation + building time of D register
Now the delay is 120ns. How to shorting it is main question of parallel acceleration of AES. The delay of Encryption a block data is the delays of global clock multiply the number of round.

- How to established pipeline

Pipeline is a common way in parallel acceleration. The mul-round crypher structure of AES facilited the establishment of pipeline. And as the key can be independently extension, the pipeline can be established using the Harvard architecture.

Step1: once extend all keys at first;
Step2: regard every round of encryption as a step of pipeline.At the end of every step, result and key exclusive or;
Step3: use D register between steps;
Step4: all D register uses the same global clock.

At the same time, we use RTL (Register Transfer Link) to produce a control chain. The length of chain is equal to the number of step of pipeline. Because of in department of the data and control, in every step we can get a result of a step. So result of a block of data may be got in every clock cycle.

Establishing pipeline require much resource, especially the register resource. If resource permit, we can construct the pipeline as we has discussed. If there is no plenty of resource, we can treat several round ciphers as a single step.

Two methods can be considered: One is that the several rounds of crypting are processed asynchronously, so that the register resource can be saved greatly. It has little effect on the performance of algorithm (possibly it may improve the performance). But it increases the competition and risk, and has effect on the stability of algorithm and the whole thermal of FPGA. Another is that the only one round cipher is still asynchronously computed in one step. But constructing several times self-feedback to complete several round cipher. It will save all kinds of resource, and not increase the competition and risk. But it will affect the cipher speed, and control chain must construct the self-feedback mechanism relatively.

As the development of FPGA, the latter is practicable. And in the end part of this paper, we'll see because of the existence of other speed bottleneck, the speed of the latter should meet the request.

5. PARALLEL ACCELERATION BASED ON XINLINX'S SPARTANII STRUCTURE FPGA

From the previous discussion, we can know: great quantity of compute can transfer to the search in the table of range of 2^8. It will fit the table searching FPGA structure of Xilinx. The in chip logic resource of FPGA of Xilinx is rich, it make the establishment of pipeline possible. The routing resource of Xilinx FPGA is rich, it is helpful of adjust the delay among the parallel task, and can easy gain the task overload balance. Otherwise based on the special structure of SPARTANII, there are several special aid parallel acceleration ways.

- BLOCK RAM

Block RAM in FPGA of SPARTANII, as the following list:

Device	Block RAM （bits）
XC2S15	16 348
XC2S30	24 576
XC2S50	32 768
XC2S100	40 960
XC2S150	49 152

As we have discussed, the save of round const and once key extension need RAM sized 1580 bits. Every type of SPARTANII FPGA can meet the request. RAM is uniformly distributed in the round of every CLB of FPGA. And using copper working procedure provides two group I/O. So it can combine pipeline and the distribution inside the FPGA, adjust the round const and extended key saved location, short the routing distance, reduce the line delay and make it possible to construct pipeline inside the FPGA. It is possible to construct pipeline inside FPGA.

The other important use of block RAM is constructs the I/O part of the algorithm. To make pipeline available, I/O must match the cipher speed. Using DMA is a good method, in theory constructing AES pipeline have 14 rounds at the most, every round can compute 256 bits at the most, It need that block RAM's size is 3584 at the total. So it will get the first result after input the last block. It can immediately output the compute result of the first block and transport back to back, this means can parallel the execution of algorithm and I/O.

- DLL(Delay Lock-Loop)

As the recent technology, using the lower outside clock will facilitate the stability of FPGA (In the DEMO board of our lab, we only use the oscillator of 25MHz). But in the establishment of pipeline, especially the working procedure is divided subtle and it request different locked loop, and it request to keep the ascend border consistent. Because our lab is concern on the work of the fine the pipeline of AES, now can't provide the concrete method about how to reasonably utilize DLL to fine the pipeline of AES.

6. THE RECENTLY RESEARCH PHASE AND THE SPEED ESTIMATED

We have finished base development of implement and parallel acceleration of AES using FPGA. Now pipeline is establishing and fractionizing. Some result have been got:

	DELAY (ns)
A round of Encrypt asynchronous	110
A block of Encrypt（12 rounds）	1440
A round of key . expansion synchronous	70

The speed of AES should be 133Mbits/S. Because pipeline establishing and fractionizing have not been finished, it is possible that the speed will greatly increase after they are finished. The estimated result will be 700Mbits/S。

At the same time, we notice that now the standard of I/O are the other bottleneck of speed. Bus of 32bits is still mainly using. For example, the bi-directional I/O speed of PCI2.1 is only 528 Mbits/S（when in DMA）. It is lower than the speed of AES. How to increase the speed of I/O is a new question of parallel acceleration of AES.

REFERENCES

[1] AES home page: http://www.nist.gov/encryption/aes

[2] Joan Daemen, Vincent Rijnmen, "AES Proposal: Rijndael", dated September 3 1999

[3] Joan Daemen, V. Rijnmen, "Answer to 'New Observations on Rijndael'", August 11, 2000

[4] Brian Gladman, "AES Second Round Implementation Experience", January 30th, 2000

[5] Brian Gladman, "The AES Algorithm (Rijndael) in C and C++", October 10th, 2000

[6] XLINX Data Book 2000, 《The Programmable Logic》

[7] 张超、陆浪如 等, "AES算法的代码分析与快速实现", 《交通与计算机》2001. 第二期.

Implementation of SMEs CA Based on Windows 2000

Meng Bo
Computer Science& Technology Department, Wuhan University of Technology
Wuhan, Hubei 430063 China
Email:tete@263.net

Xiong QianXing
Computer Science& Technology Department, Wuhan University of Technology
Wuhan, Hubei 430063, China
Email: QXXI@public.wh.hb.cn

ABSTRACT

The issuance of windows 2000 makes small/middle enterprises (SMEs) develops its independent Certificate Authorities (CA) possible. Certificate Authority is a core element of electronic business security applications based on Public Key Infrastructure (PKI). This paper discusses the structure of Public Key Infrastructure and analyzes Windows 2000 Public Key Infrastructure and certificate services. The paper puts out a method how to establish small/middle enterprises Certificate Authorities based on Windows 2000 PKI and its certificate services and gives an implementation.

Keywords: Public Key Infrastructure, Certificate Authority, Policy Module, Electronic Business, Certificate Services

1. INTRODUCTION

With the popularity of electronic business, the requirement of security is getting more and more imperious. Although there are many definitions on security, people have the same idea on its core concepts, such as information confidentiality, authenticity, information integration, access control and no-repudiation. The key question of development application system of electronic business is how to use the security technology to establish the Internet security structure. Now the Certificate Authority system based on Public Key Infrastructure can achieve the demands of information and network security. The core of Certificate Authority system is Certificate Authority. Certificate Authority is the fair third party and establishes the foundation of the authority framework of authentication. Microsoft puts out windows 2000 in 2000 that includes the Windows NT certificate services and makes it possible for SMEs to develop its Certificate Authority.

2. PUBLIC KEY INFRASTRUCTURE

Public Key Infrastructure (PKI), which conforms to the international standards and is key management platform, can provide the services such as key and certificates management which are needed by network application based on encryption, decryption and digital signatures and so on. Public Key Infrastructure must include Certificate Authority, certificates repository, key backup and recovery system, certificates revocation system, PKI application interface system etc.

Certificate Authority

A certificate authority is simply an entity or service that issues certificates and is core of PKI.

A certificate is a particular type of digitally signed statement. We can use it to verify the validity of an entity identity and its public key. In the network environment, which applied the public key cryptography, we must verify the validity of an entity identity and its public key. So there is a trusted organization to verify the validity of an entity identity and its public key. Certificate Authority is the organization. A CA acts as a guarantor of the binding between the subject public key and the subject identity information contained within the certificated it issues. Different CAs may choose to verify that binding via different means, so it is important to understand the authority's policies and procedures before choosing to trust that authority to vouch for public keys

Certificates Repository

Certificates were conserved in certificates repository that is public information repository. User can get other user' public key and certificate. The method of implementation of certificates repository is that we can use catalog system supported Lightweight Directory Access Protocol (LDAP). Users access the certificates repository by LDAP. At the same time system must maintain the integrality of certificates repository.

Key Backup and Recovery System

if user lost its private key, encrypted data can not be decrypted and is not used by user. In order to avoid the case, PKI must provide key backup and recovery mechanism. Key backup and recovery is performed by the trusted organization. such as CA.

Certificate Revocation System

Certificate revocation system is a important component of PKI. Certificates tend to be long-lived credentials and there are a number of reasons why these credentials may become untrustworthy prior to their expiration. PKI must provide certificate revocation mechanism.

PKI Application Interface System

The value of PKI is services such as encryption, digital signature that are used by user. So PKI must provide application interface system that makes all kinds of

applications interactive with PKI on the security, consistent and trusted mode.

3. WINDOWS2000 PKI COMPONENT

Windows 2000 introduces a comprehensive public key infrastructure (PKI) to the windows platform. This extends the windows public key cryptographic services, providing an integrated set of services and administrative tools for creating, deploying, and managing PK-based application. This allows application developers to take advantage of Windows NT, s shared-secret security mechanism or PK-based security mechanism as appropriate.

Figure 1 presents a top-level view of the components that make up the windows 2000 PKI. This is a logical view and does not imply physical requirements for separate servers; a key element in the PKI is Microsoft certificate services. These CAs support certificate issuances and revocation. They are integrated with Active Directory, which provides CA location information and CA policy, and allows certificates and revocation to be published.

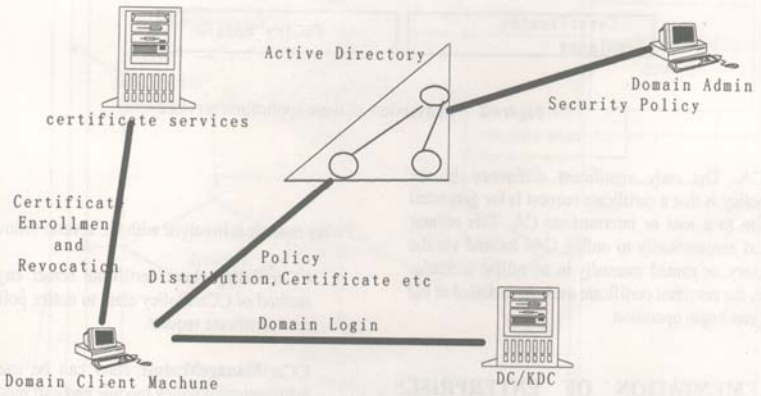


Figure 1 windows 2000 public key infrastructure

The PKI does not replace the existing Windows NT domain trust and authorization mechanisms based on the domain controller and Kerberos Key Distribution Center. The PKI works with these services and provides enhancements allowing applications to readily scalable and distributed identification and authentication, integrity, and confidentiality.

Support for creating, deploying, and managing PK-based applications id provided uniformly on workstations and application servers running Windows NT as well as workstation running Windows 95 and windows 98.

Figure 2 provides an overview of these application services.

Microsoft CryptoAPI is the cornerstone for these services. It provides a standard interface to cryptographic functionality supplied by installable cryptographic providers (CSPs). These CSPs may be software based or take advantage of cryptographic hardware devices, and can support a Varity of algorithms and key strengths.

Certificate Services

Microsoft Certificate services, included with windows NT 5.0 server, provides a means for an Enterprise to easily establish CAs in support of their business needs. Certificate Services include default policy modules suitable for issuing certificates to Enterprise entities. This includes identification of the requesting entity and validation that the certificate requested is allowed under the domain PK security policy. This may be easily modified or enhanced to address other policy considerations or to extend CA support for various extranet or Internet scenarios. It provides broad support for PK-enabled applications in heterogeneous environments.

Deploying Microsoft Certificate Services is a fairly straightforward operation. It is recommended that you establish the domain prior to creating a CA. Then establish an enterprise root CA, or CAs. The Certificate Services installation process walks the administrator through these process key elements in this process include:

- Selecting the host server
- Naming
- Key generation
- CA certificate
- Active Directory integration
- Issuing policy

After a root CA has been established, it is possible to install intermediate or issuing CAs subordinate to

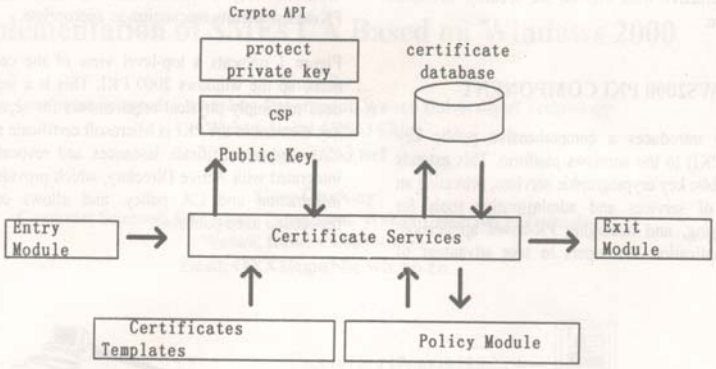


Figure 2 An overview of these application services

This root CA. The only significant difference in the installation policy is that a certificate request is for generated for submission to a root or intermediate CA. This request may be routed automatically to online CAs located via the Active Directory, or routed manually in an offline scenario. In either case, the resultant certificate must be installed at the CA before it can begin operation.

4. IMPLEMENTATION OF ENTERPRISES CA

Enterprises can take advantage of Microsoft certificate service to establish CA. Before deploying CA, we need to define certificate policy and certificate process specification (CSP). Certificate policy describes which aspect certificate can be applied and the responsibilities for applying the certificate. CPS defines how to implement the certificate policy in the operation policy, system structure, physical security, and computing environments that CA organizes.

In the following we describe the implementation of enterprise-customized module.

Applying windows 2000 PKI and certificate services component to establish CA needs a enterprise policy module which implements certificate policy by using visual basic 6.0.

A CCertPolicy class is needed. Policy module can call the method of CCertServerPolicy class to deal with certificate requests of users. CCertServerPolicy class can read and write certificate properties. It needs one more CCertManageModule class in windows 2000 than in Windows NT.

In order to debug the enterprise customize module, certificate services should be in console mode. First use "net stop certsvr" to stop certificate services. Then execute "certsrv –z". And the output of certificate services is on the screen. "ctrl+c" can exit certificate services.

Policy module is involved with the several following classes:

CCertPolicy class: certificate server engine call the method of CCertPolicy class to notice policy module a new certificate request.

CCertManageModule class can be used to extract information of policy module and exit module.

CCertServerPolicy class can be used to collect data item of certificate, set and modify certificate property.

Figure 3 describes the flow of dealing with the certificate application. When a new certificate application arrives, policy module can collect information of all kinds of property by the method of CCertPolicy class, then CCertManageModule class set policy information, at this time CCertPolicy class can decide if certificate application is eligible. If it is eligible, it need to be decided weather it should be submitted to administrator or not according to rules; if it is not eligible, it also need to be decided weather to deny it directly or not by the denying means. Policy module can deal with certificate application by one of three kinds of means. In the last policy module return the result to the certificate services engine.

5. CONCLUSION

With the development of electronic business application, a lot of SMEs want to establish it selves independent CA. The issuance of windows 2000 makes it possible. Windows 2000 introduces a comprehensive public key infrastructure (PKI) to the windows platform. This extends the windows public key cryptographic services, providing an integrated set of services and administrative tools for application. So it is to easy for SMEs to develop its independent CAs. The paper discusses the structure of windows 2000 PKI and certificate services .At the same time it puts out a example.

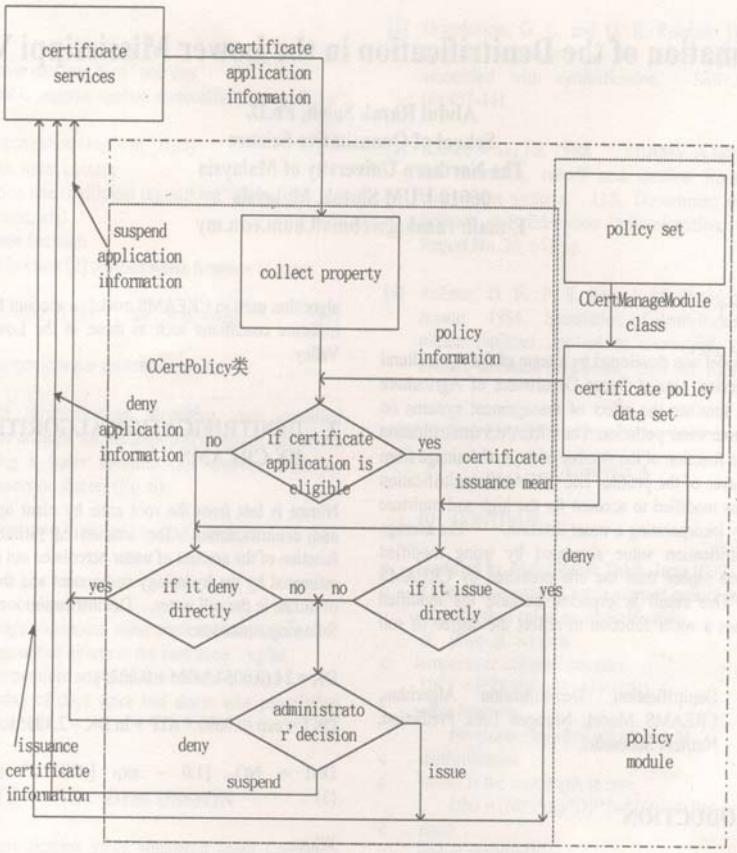


Figure 3 the flow that policy module deals with the certificates

REFERENCES

[1] Microsoft. Step-by-Step Guide to Setting up a Certificate Authority. http://www.Microsoft.com.2000.2
[2] Microsoft.Step-by-StepGuide to Administering Certificate Services. http://www.Microsoft.com.2000.2
[3] RSA Laboratories. Public Key Cryptography Standards. http://www.rsasecurity.com.2001.3
[4] Microsoft.MSDN Online. http://www.Microsoft.com. 2000.10. Platform SDK

Estimation of the Denitrification in the Lower Mississippi Valley

Abdul Razak Saleh, Ph.D.
School of Quantitative Science
The Northern University of Malaysia
06010 UUM Sintok, Malaysia
E-mail: razak@webmail.uum.edu.my

ABSTRACT

CREAMS model was developed by a team of the Agricultural Research Service, United States Department of Agriculture scientists to simulate the effect of management systems on nonpoint source water pollution. The CREAMS denitrification algorithm is a function of the number of days of drainage from the bottom layer of the profile. The CREAMS denitrification algorithm was modified to account for the high soil moisture conditions by incorporating a water function. The average annual denitrification value simulated by using modified algorithm was higher than the one predicted by CREAMS algorithm. This result is expected because the modified algorithm uses a water function to reflect the degree of soil saturation.

Keywords: Denitrification, Denitrification Algorithm, CREAMS Model, Nitrogen Loss Prediction, Nutrient Submodel.

1. INTRODUCTION

Nutrients are naturally occurring chemicals essential for plant growth. A total of 16 chemical elements are necessary for the growth and reproduction of most plants, although the most significant are nitrogen (N), phosphorus (P) and potassium (K). Most soils are deficient in N, P, and K, and thus chemical fertilizers are added to the soil for optimum plant production. These chemicals may become pollutants when they are transported away from their place of application. Present evidence indicates that nitrogen and phosphorus are the principal nutrient pollutants from agricultural lands [1].

Rising concerns regarding the effects of agricultural management practices on the environment and the increasing capabilities of computers, have prompted the development of complex mathematical models for evaluating such effects. CREAMS (Chemical, runoff, and Erosion from Agricultural Management Systems) model was developed by a team of the Agricultural Research Service, United States Department of Agriculture scientists to simulate the effect of management systems on nonpoint source water pollution [3] .The model consists of three components, which describe field hydrology, erosion and sedimentation, and chemistry.

The hydrology component estimates runoff volume and peak rate, infiltration, evapotranspiration, soil water content, and percolation on a daily basis. The erosion component estimates erosion and sediment yield including particle distribution at the edge of the field on a daily basis. The chemistry component include elements for plant nutrients and pesticides.

The objective of the study was to modify the denitrification algorithm used in CREAMS model to account for the high soil moisture conditions such as those in the Lower Mississippi Valley.

2. DENITRIFICATION ALGORITHM USED BY CREAMS

Nitrate is lost from the root zone by plant uptake, leaching, and denitrification. The amount of nitrate leached is a function of the amount of water percolated out of the root zone estimated by the hydrology component and the concentration of nitrate in the soil water. Denitrification is estimated by the following equations:

$$DK = 24\,(0.0064 * OM + 0.0025) \quad (1)$$

$$DKT = \exp(0.0693 * ATP + \ln DK - 2.4255) \quad (2)$$

$$DNI = NO_3\,\{1.0 - \exp[-DKT * (DT - 0.5)]\} \quad (3)$$

Where,

- DNI = denitrification between storms (kg/ha),
- OM = organic matter (%),
- DK = the rate constant at 35^{0}C/day,
- DK = temperature adjusted rate constant,
- ATP = average temperature (^{0}C),
- NO = amount of nitrate in the root zone (kg/ha), and
- DT = drainage since the last storm (day).

3. MATERIAL AND METHODS

The CREAMS denitrification algorithm is a function of the number of days of drainage from the bottom layer of the profile. The assumption is that while drainage is occurring and the soil profile moisture content is above field capacity, anaerobic conditions persist in the soil thus enabling denitrification [3]. This assumption, for the shallow water table soils such as those in the Lower Mississippi Valley, will not give satisfactory soil moisture estimates. The CREAMS model overestimates nitrogen loss due to underestimating the denitrification because the model underestimate soil moisture.

The CREAMS denitrification algorithm was modified to account for the high soil moisture conditions by incorporating a water function. Rolston et al. [4] proposed a simulation model in which the denitrification rate was considered to be a function of nitrate concentration, water-extractable organic carbon concentration, degree of soil water saturation, and temperature given by:

$$F = k\theta f_w f_T CN \quad (4)$$

Where,
F = denitrification rate (mg N m^{-3} soil day^{-1})
C = water-soluble organic carbon concentration (mg C kg^{-1} soil)
N = nitrate concentration (mg N m^{-3} H_2O)
θ = volumetric water content
k = denitrification rate coefficient (kg soil mg^{-1} C day^{-1})
f_w = water function, and
f_T = temperature function
Grundman and Rolston [2] defined water function (f_w) as:

$$f_w = \{[(\theta/\theta_s) - 0.62]/0.38\}^{1.74} \quad \text{for} \quad \theta/\theta_s > 0.62 \qquad (5)$$

where θ_s is volumetric water content at saturation.

The CREAMS denitrification algorithm was modified (Appendix A) to account for the high soil moisture conditions by incorporating a water function (f_w) together with the appropriate conversion factors (Eq. 6).

$$DNI = NO_3 * DKT * DP * f_w * 100000.0/(RZMAX * AWC \qquad (6)$$

where,
RZMAX = root depth (mm),
AWC= average volumetric water content (mm^3/mm^3),
NO_3 = amount of nitrate in the root zone kg/ha,
DKT = temperature adjusted rate constant, and
DP = number of days since last storm whe percolation Occurred

4. RESULTS AND DISCUSSION

The annual denitrification value simulated using CREAMS algorithm and modified algorithm are presented in Table 1. The average annual denitrification value simulated by using modified algorithm was higher than the one predicted by CREAMS algorithm. This result is expected because the modified algorithm uses a water function to reflect the degree of soil saturation. The CREAMS algorithm uses the number of days of percolation which is the false indicator of degree of saturation.

Table 1. Simulated Yearly Total Denitrification (kg/ha)

Year	CREAMS algorithm	Modified algorithm
1981	301.02	341.11
1982	255.48	271.01
1983	211.26	268.42
1984	270.13	265.60
1985	260.08	278.39
1986	197.86	234.48
1987	223.92	280.29
Average	245.68	277.14

REFERENCES

[1] Frere, M.H., J. D. Ross, and L. J. Lane. 1980. The nutrient submodel, In:Knisel W. G., (ed.), CREAMS: A field-scale model for chemicals, runoff and erosion from agricultural management systems. USDA Cons. Res. Rep. No. 26, 643 p.

[2] Grundmann, G. L. and D. E. Rolston. 1987. A water function approximation to degree of anaerobiosis associated with denitrification. Soil Science 144 (6):437-441.

[3] Knisel, W.G., Ed. 1980. CREAMS: A field scale model for chemicals, runoff and erosion from agricultural mangement systems. U.S. Department of agriculture, Science and Education Administration, Conservation Report No. 26, 643 pp.

[4] Rolston, D. E., P. S. Rao, J. M. Davidson and R. E. Jessup. 1984. Simulation of denitrification losses of nitrate fertilizer applied to uncropped, cropped, and manure-amended field plots. Soil Sci. 137(4):270-279.

Appendix A

```
      40  CONTINUE
c     Modified by Abdul Razak Saleh, June 1992
c     if average volumetric water content greater than field
      capacity, then calculate denitrification
       if  (awc .gt. fc) then
c     temperature adjusted constant
       DKT = EXP (0.0693*ATP+DB)
c     water function
          fw=(((awc/por)-0.62)/0.62)**1.74
c     denitrification
c     rzmax is the root depth in mm
          DNI = (N03*dkt*DP*fw*100000.0)/(awc*rzmax
c     endif
c     end of modification

      70  CONTINUE
```

Design and Implementation of CAI Revision-type Question-base

Jiang, Xiaoyao
Department of Computer Science & Technology
Anqing Normal College
Anqing, 246011, Anhui, P. R. China
Email: jiangxy@aqtc.edu

ABSTRACT

Design and implementation of a learning system of web-based CAI (Computer Assisted Instruction) revision-type question-base is reported in this paper. The system can provide users a remote education environment, in which the CAI question-base can be used to learn, revise and test through the computer network.

Keywords: CAI Test-base, Test system, Web Technology, ASP (Active Server Pages), Remote Education

1. INTRODUCTION

The development of the technologies of computer network has greatly promoted the reform of the education techniques and modes. Remote education based on the web is very prospective in its development and application.

Revision-type question-base system is mainly used for learning, testing and grading for a given subject. With the network platform, a remote education and learning system can be established. This system is helpful for learners to effectively make use of the education resources; and clearly it is a new mode of education and learning which deserves extensive investigation.

2. ANALYSIS AND DESIGN OF THE SYSTEM

1) Objective of system design

The objective of this system design is mainly to implement the web-based revision-type question-base learning and testing system.
This system can be used to learn the new knowledge, to revise, to test and grade and to diagnose, either locally or remotely. It is convenient for students to use this system in effectively learning and understanding the new knowledge; and the system is also helpful to keep the students informed of the status of their learning.

2) Function and module of the system

This system is composed of two parts.

The first part is the module used to manage the question-base. This module fulfils collecting the questions and forming the question-base; browsing, maintaining and renewing the question-base; and forming the test-sheet. The first part is carried out by the Web-Server and the Database-Server.

The second part includes the revision module and the test module. After the system verifies the identity of the student, it provides the student with the interface containing the main points of the subject and the revision contents according to the privilege of the student. And the second part also provides tests to the student, and gives out its evaluation and feedback according to the learning procedure and test results. The second part is carried out by the User-Terminal and the Web –Server.

3. IMPLEMENTATION TECHNOLOGY OF THE SYSTEM

1) The structure of the system

The system uses 3-tiered architecture. The lowest tier is the client based on the Web-Browser, which is used for students to communicate with computers, in verifying the identity, displaying the data, and providing feedback information. The middle tier is the Application-Server; it can process a request to access and to communicate with the Database-Server. The top tier is the Database-Server; it can process a request of accessing, updating, and maintaining the Question-Base. Comparing with the traditional Client/ Server architecture, it has the advantages as follow.

① Less dependence of the User-Terminal on the Database-Server. This makes it easy to transplant the User-Terminal.
② Higher quality of service on the Web-Server. Since all users access data only through Web-Server, the database is effectively protected from the potential improper accessing.
③ Users do not have to maintain the data of the question-base. The maintenance of the question-base can be carried out on the Web-Server and the Database-Server; this is the so-called "fat server and thin user" mode. This mode greatly enhances the availability of the system resources, and ensures openness and expandability of the system.
④ Applications are executed on the server, and the browsers only receive results returned from the server. The efficiency of the system is consequently improved.

2) Main techniques used in the system.

① ASP techniques
ASP techniques can operate complex databases according to users' requirements, forming the interactive pages. This system mainly uses ADO (Active Data Object) accessing

database techniques to store and access the database for the Web pages. Through ODBC (Open Database Connectivity), ADO writes, reads and operates the data in the database, providing information to students, and helping browser-users to visit the web database by SQL (Structure Query Language) command.

The procedure of visiting the web database through ASP is shown in Figure 1.

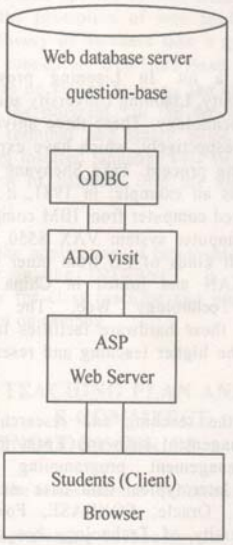


Fig.1 The procedure of visiting the web database through ASP.

Students' user-ends use the IE (Internet Explor) techniques to visit Web database and question-base through IE server and intermediary. Revision-type question base software is put on the Web server as an intermediary, providing the communication and application services.

② HTML (HyperText Markup Language)
HTML has international standard, and it can be used across platforms. It is easy to learn this language. And this language is strong in supporting super-media, convenient in constructing complex information network of super-media, and compatible in using many different formats of files. FrontPage is used in this system, combining with VB Script and Java Applet tools.

③ Development tools of Delphi database
It is easy to visit database by using Delphi to construct Web server. One can get data needed from the data sets with the aid of some standard files, and form the active HTML FrontPage.
Delphi supports remote database visiting and SQL consultation, and its database components supply an object-oriented interface for the development of applications.

3). Operation environment

Hardware environment: Web server, Database server, User-terminal, Exchanger and Net-card.

System platform: the browse/server system structure; the Windows NT4.0 network operation system for the server; IIS3.0/IIS4.0, Delphi database as software for Web server; Windows98, Internet Explorer, SQL, Server on the user-terminal.

Development Tools: Delphi 5.0; FrontPage; VB Script, etc.

Web-based remote education is a new mode of education and a new mode of learning. And the web-based learning software has been of wide interest in network application research community. The system reported in this paper has been successfully applied in the course of Physical Chemistry for Chemistry Discipline and the course of Data Structure for Computer Discipline; it has provided students a local and remote environment of learning. The preliminary results of these applications of the system are encouraging.

* Partially supported by Anhui education official grant ZD99001.

REFERENCES

[1] XU XingHua, Delphi 5 Advanced programming ---- COM, CORBA, and Internet programming, People's Post and Telecommunications Press. Pub, MAY 2000 ,pp.197-391
[2] Myung Geun LEE, Profiling Students' adaptation styles in Web-based learning, Computer and Education Vol.36,Feb.2001,pp.121-131
[3] Wang guorong, Active Server Pages & Database, People's Post and Telecommunications Press. Pub March 1999, pp.15-134
[4] xue yuanjun, gujiayin ,et al, Publication and Design of Database on Network, Tsinghua Press. PUB August 1999, pp.27-159

Teaching and Research on the Series of Courses of E-commerce Design Technology

Zhang Jianhua
School Of Information Science and Engineering,
Shenyang University of Technology,
Shenyang, Liaoning, China
Email: shanliangrensheng@sy163.net

ABSTRACT

This article illustrates the author's ideas and his understanding on the course setup, teaching plan and teaching methods of the series of courses of e-commerce design technology in the nation-wide universities of technology.

Keywords: E-commerce Dynamic Web site Teaching and Research.

1. TRANSFORMATION OF EDUCATION SHOULD MEET THE NEED OF SOCIAL DEVELOPMENT

With the development of computer science and technology, INTERNET and multi-media technology have become a creative means to widen human abilities. At present, faced with the entrance of WTO in China, it is urgent to carry out "e-commerce and trade" in enterprises and institutions. In fact, in the functioning technique of INTERNET, one of the most important tasks is to carry out e-commerce and to make it practical. However, the knowledge concerning "the relationship between the undertaking of e-commerce in net", the computer and network technique is not so popular. We can even say that it's just recent one or two years that we start to open this course in universities. It's our duty to cultivate this kind of talents for society vigorously.

The opening of the series on "e-commerce design technology", especially the opening on a great scale for the non-major computer students in general science and engineering universities, is worthy of being considered deeply by us educators in higher learning institution. The followings are my ideas about the series of courses above.

2. THE UNIVERSITIES OF TECHNOLOGY ARE QUALIFIED FOR THE OPENING OF THE SERIES ON "E-COMMERCE DESIGN TECHNOLOGY"

2.1 The Universities have been well qualified in the hardware surroundings

In the late 70s, universities of technology in China have opened the specialty in computer application or related subjects, and have begun to undertake teaching and research. In the past twelve years, with the development of science and technology, the teaching conditions have been improved a lot. In Liaoning province, like Northeast University, Liaoning University and Shenyang University of Technology, These three universities are distinctive and respectively, which have experienced a similar developing process. Take Shenyang University of Technology as an example: in 1981, it introduced Series/1 mini-sized computer from IBM company. Then it introduced computer system VAX 8550 from DEC company, and all kinds of PC from other companies. They formed LAN and joined in China Education Science and Technology Web. The increasing improvement of these hardware facilities laid a good foundation for the higher teaching and research in the university.

Let's look at the teaching and research for the commercial management software. From the earlier commercial management programming language COBOL to the later typical data base management system DBASE, Oracle, FOXBASE, FoxPro etc, Shenyang University of Technology has conducted corresponding research and application and also taught several related courses.

2.2 The undergraduates have generally acknowledged the fundamental knowledge for the computer major

Recently, with the further transformation of education, the teaching syllabus and schedule in universities have been improved a lot in order to adjust to the high-speed development in social economy and the fact that society needs talents eagerly. Each university has also done a lot of beneficial work, such as opening optional courses, giving lectures on a special topic. And meanwhile, it has achieved great progress in it. One of the achievements is that their vision has been widened. At present, the undergraduates majoring in different specialties have all qualified with the basic knowledge related to computer in the universities. For example, they can run operating system WINDOWS98 freely. They have learned PASCAL, C or other programming language. They have learned one or two type of DBMS too, and so on. These courses have laid a certain foundation for the opening of a series of web application courses (programming basis, Database application basis, communication and network technique basis, multi-media application basis, etc).

3. THE OPENING OF THE SERIES OF COURSES IN E-COMMERCE DESIGNING TECHNOLOGY EMBODIES THE ENHANCEMENT OF THEORETICAL LEVEL IN THE RELATED COURSES

The opening of the series of courses in e-commerce designing technology is not only a simple change in the teaching content of related courses, but also a symbol in the enhancement of theoretical level. Take program designing as an example, the "Object-Oriented program designing" is most popular designing form at present, which removes the traditional concepts and produces a radical change in the ways of analyzing problems, designing and even thinking. Another example, the research for data base: the database based on distributed system has been applied to web field, which not only improves the promotion of web application, but also helps the theory of database take a great leap—giving rise to the concept of Web Data Base: "It can store the date and can be visited by using the inquiring language or editing the program in the server/client system's customers terminal and return some information for them. Using inserted language PHP or ASP can putting the function of database into the applying programs through CGI."

As far as the application of communication and Web technology in the INTERNET, it offers a good opportunity for class teaching and practical teaching combination on campus.

4. THE TEACHING PLAN AND PRACTICE OF E-COMMERCE DESIGN TECHNOLOGY SERIES

I think that the series include the following courses:

4.1 The Programming Technology for Web Page

It includes:

1) HTML (Hyper Text Markup Language)
   It is the base of designing web page.
2) CSS (Cascading Style Sheets)
   It not only makes up the shortcoming of HTML, but also makes itself even more flexibly by using script programming.
3) Application of DOM (Document Object Model) in HTML
   It is a new concept in programming and simplifies the work greatly.
4) Use of Script
   Control the subject in the Web page and the information can be visit interactively.

Such as above, It is the base of Web page designing; which is necessary to establish for the DHTML designing.

In our university, it is an optional course for sophomores and finishes it in 28 to 32 class hours We have three understandings in the class teaching: First, the practice is very important, the students do exercises by themselves in class is important too. Second, presents demonstrative questions as many as possible. Third, enlarge the knowledge by introducing related software, for example, getting the files of picture (or video, audio etc), and transferring their format each other.

4.2 Management of Dynamic Web Site

The course includes:

1) Choice of soft and hardware.
2) Establish the site.
3) Safety and management of the site.

4.3 Application of Web Database and E—commerce Design

It includes:

1) PHP (Personal Home Page)
   It's the base of the DHTML designing in the Server /Client systems. This programming language is inserted in HTML.

   Note: Another choice is ASP (Active Server Page) or others.

2) Application of My SQL in the web site
   It's the key point of teaching.

Concerning that the students of science and engineering universities in our country have learned the database, this course will focus on the basic operation and application. For example, after introducing the mode of database, the distinction of distributive database and centralized one, the main task is to introduce the basic operation of My SQL (Create, Open, Delete, Inquire and paragraph operation etc). And the program design should be made along the line of classic e—commerce cases.
Note: Other DBMS may be chosen (for example: MS SQL Server).

In urgent need of teaching reform, we suggest that the two courses "Management of Dynamic Web Site" and "Application of Web Database and E — commerce Design" are combined as one. The suitable name is "Dynamic Web Site and E — commerce Design Technology." The teaching should be focused on database application. The plan class hour is 32 or more.

4.4 Application of classic Web page design software

This software include:

Macromedia Flash 5, the solution for professional Web animation design and production.
Macromedia Fireworks 4, the solution for professional Web graphic design and production.
Macromedia Dreamweavers 4, the solution for professional site design and production.

Because the students of advanced universities have self-teaching ability, this course can also be used to be self-learnt course. Or guides them in the web page design competition or program design exchange.

It is believed that strengthening practice is the important condition of improving the teaching quality. They should get enough opportunity of go to the society and take part in the relative practical work.

5. THE CHARACTERISTICS AND FORCAST OF E−COMMERCE DESIGN TECHNOLOGY

With the rapid progress of this technology, the teaching content should enrich gradually. For example, the standard of XHTML has been published last year, which should replace the former HTML.

In the future, the new standard of Web page and technology will develop even faster. We should keep close attention to the trends and choose the promising research subject in the teaching course.

The Application of Distributed Computing in Higher Education

Zhang Wenhua, Zhao Sanquan
Wuhan University of Technology
(430070, wuhan, Hubei, China)
E-mail: zwhlgd@263.net

ABSTRACT

The paper introduces the application of distributed computing to distance education and digital libraries in higher education, lays stress on the use and techniques of digital libraries, and also introduces the relevant situation in China.

Keywords: Distributed Computing, Digital Libraries, Distance Education, High Education

1. INTRODUCTION

When we come to Distributed Computing, we refer to not just one computer, but a net of computer. A computer has two functions. First, it can store information, which can also be called data. Second, it has the ability to deal with data, it can calculate. According to this classification, a computer does two kinds of work, Distributed Data Management and Distributed Computing.

When distributed data are deposited, the net makes them distributing, we put them in different computers in the net. And in the condition of distributed computing, data management is completed by the cooperation of several computers. For example, there is a task named P to be disposed, and it is build up by PA and PB. So we can use computer A to dispose PA, and use computer B to dispose PB. Then a distributed computing formed.

Why we use distributed computing? First, to improve the speed of disposing by sharing the resource; Second, to lighten the burthen of the net. In Distributed Data Management, we just need to transfer the result. Third, for security. We can put the most important process and data on the server. Four, in some condition, it is easier to dispose some down-to-earth problems.

The computer art is developing form the original large host computer structure into distributed? Computing oriented client/servers based on local area network. Telnet/Remote login also supports Connection (terminal to terminal) and distributed computing (process to process) communication.

As a computational algorithm, distributed computing allows application program to run in the same way over different types of network. The environment of distributed computing was put forward by the Open Software Foundation (OSF). It is a series of standards laid down for servers, interfaces and protocols that can conduct distributed computing.
To play well the part of libraries of libraries in the society of knowledge economy, all the countries in the world are making use of Internet to develop distance education and prepare digital Libraries.

2. THE APPLICATION OF DISTRIBUTED COMPUTING IN HIGHER EDUCATION

America took the lead in studying DL in 1991.China formally carried out the engineering project of National DC in Aug.1998.Chinese Pilot Digital Library passed technical appraisement n May 30.2001.Thus a distributed, extensible and interoperable DL frame with a certain a scale of content resource has been founded in China.

2.1 The application of distributed computing to Chinese distance education

Different from Chinese electrical business affairs' vigor and vitality, the network universities opened in silence. After all, Chinese distance education has made big strides forward in 2000. Till now 31 universities have been approved as experimental units of modern distance education by the Ministry of Education The Modern Distance Education Cooperation Group of Colleges and Universities has been set up in Peking so as to strengthen interchange and cooperation and promote the building and share of teaching resources.

In September 1999,the item of the high-speed main line network was set up, which goal is to achieve Chinese Education and Research Net (CERNET) before the end of December 2000[2]. By means of the platform, television, multimedia and dozens of other programs can be transferred simultaneously at different rates. The satellite Internet that was launched by the system has joined in the service to tie in satellite Internet with ground CERNET, thus the satellite communication network and the optical fiber communication network have been amalgamated into the educational instruction network with bi-directional interactive function .It has thoroughly changed the condition that Chinese satellite television only could transmit one-way television programs. According to the latest relevant statistical figures from the Chinese Ministry of Education, the 31 experimental universities have a register of about 19,000 students. Most of who study to acquire the record of formal schooling. The system of continuous education is also structured [3]. The concrete expression of advanced communication technique in remote education is the digital network, the digitalization of teaching media and the multimedia of teaching information. The remote education urgently need digital library to supply the guarantee of document and knowledge information.

2.2 The application of distributed computing to digital library

America took the lead in studying DL in 1991. China formally carried out the engineering project of National DC in Aug. 1998.Chinese Pilot Digital Library passed technical appraisement on May 30, 2001. Thus a distributed, extensible and interoperable Dl frame with a certain a scale of content resource has been founded in China [4]

2.2.1 The concept of the digital library (DL)

The DL is not just composed of the automatic system and library stock of one certain library. In reality, a future digital library cannot be measured merely by the amount of it's book or magazine stock, the built data base scale or automatic system, but by the quantity and quality of the information supplied to its readers after utilizing the network resource for quick inquiry, i.e., make use of the most sophisticated computer equipments and technology to supply the most accurate, most comprehensive and most useful information to its users through high speed network by most convenient means and the quickest speed. The data resources can be those that are distributed over many libraries in different areas, in a country or all over the world. This actually is just users' expectation of resource sharing by means of network libraries or virtual libraries.

Thus it can be seen that the digital library is a vast knowledge engineering with complex technology, huge scale and rich digital resources. To realize the engineering, a great quantity of manpower, material and financial resources and quite a long time are needed [5].

2.2.2 The function of DL

The basic function of DL various digital carriers, data mining, description, storage and management, the effective access and query to distributed databases of different structures, system management and copyright protection.

2.2.3 The key technique of DL

2.2.3.1 Network technique

The aim of the Next Generation Internet (OC-48 "IP over DWDDM") supported by the American Federal Government is to study and develop more advanced network technique so as to apply it to distributed computation earlier and more reliably. The Internet 2 undertaken by up to a hundred universities and the Next Generation Internet are the complement of each other. As an advanced infrastructure, the Next Generation Internet puts particular emphasis on the enhancement of innovation level of academic research and education.

The Next Generation Internet sponsored also by colleges and universities in China is similar to the Internet 2 in America. Since it had started late, it adopted more advanced technique. Chinese high speed experimental INTERNET--NSFC net has taken effect [6].

The new three layers of distributed architecture is needed in network for DL.(display logic, Web server/service logic and data access logic).

2.2.3.2 The technique of multimedia distributed databases

DL should provide an interactive customizable interface for querying multimedia-distributed databases [7]. The technique of multimedia distributed databases should also be supported by software intelligent agent [8,9], full-text research (e.g. Chinese full-text research TRS software) and powerful meta search engine, etc..

The information retrieval technique of DL should be intellectualized and personalized. Professional knowledge and inquiry skill should be embedded in the decision-making tree and neural net [10,11].

Correspondingly, the research stress of the storage technique of multimedia distribution databases is image processing and markup language--HTML, SGML, XML [12], e.g. Chinese medicine DL presents an XML metadata design [13].

The DL Research Institute of Peking University has achieved good results in the items of overseas metadata trade-off study and Chinese Document metadata demonstration database etc. [14].

2.3 The problem of Chinese information

The simultaneous existence of the four kinds of Chinese character exchange codes has affected Chinese resource sharing. In addition, there are also three difficult problems: input mode, mode automatic identification and machine understanding, The first two have solved on the whole. The third is being studied.

3. "211 ENGINEERING" IN HIGHER EDUCATION OF CHINA

The "211 Engineering" is the broadest key constructive engineering going on in the field of higher education ever since the establishment of the People's Republic of China. The engineering's objective and task is oriented towards the 21 century. The construction centers on Chinese 100 or so key universities and a batch of key branches of learning. Up to now, 98 universities have been approved and several hundred constructive items of key courses have been arranged. Simultaneously, to heighten the integral level of higher education, the constructive item of nationwide public service system of the "211 Engineering" is also arranged, including China Education and Research Net (CER Net) and China Academic Library and Information System (CALIS).

The first phase of the construction of the "211 Engineering" will soon finish. The overall check before acceptance is going on. The building goal of the second phase is to continue the construction of the key courses and work hard for part courses

to approach or achieve advanced world standards and establish the key courses system with national overall arrangement and structure. The basic conditions as teaching, scientific research in part universities where key courses are relatively concentrated should be improved so as to quicken the construction of China's CER-NET, Digital Libraries, large-scale instrument, teaching resource sharing of postgraduate training and other public service system for further improvement on information environment, technical facility and integrated condition of running a school [15,16,17]

4. CONCLUSION

The Chinese government attaches importance to not only the reform and development of higher education defining it as the basic policy of China, but also the construction of DL, regarding it as important integrant of the national information basic installation.

We believe that the Internet has supplied a developing chance to the Chinese higher education. In turn, the "211 Engineering" of the Chinese higher educational circles will also make a contribution to the digitized China and the digitized Earth.

REFERENCES

[1] http://www.cernet.edu.cn/zhong_guo_jiao_yu/yuan_cheng_jiao_yu/ziyuan/003.php

[2] http://www.cernet.edu.cn/cernet_lanmu/cernet_jian_jie/cernetjianjie.php

[3] 王大可. 数字图书馆建设中的问题及解决办法. 现代图书情报技术，2000，16(3):10-13,15

[4] http://www.cernet.edu.cn/wang_luo_yan_jiu_yu_fa_zhan/w_cernet_dong_tai/20001130a.php

[5] Cruz, Isabel F., Lucas, Wendy T. Customizable layout-driven approach to querying digital libraries. Proceedings of SPIE—The International Society for Optical Engineering. 1999,3654. p.122-134

[6] 邢春晓，潘泉，张洪才，戴冠中。数字图书馆若干关键技术研究。西北理工大学学报，1999，17(3):419-424

[7] Chen, Hsinchum Chung, Yi-ming. Ramsey, Marshall, Smart itsy-bitsy Spider for the Web. Journal of American Society for Information Science. 1998, 49(7): 604-618

[8] A. Rauber, D. Merkl. Creating an order in distributed digital libraries by integrating independent self-organizing maps.

[9] A. Rauber, D. Merkl. Organization of distributed digital libraries: a neural network-based approach. Intelligent Data Engineering and Learning : Proceedings on financial Engineering and Data Mining.1st International Symposium. IDEAL'98, Hong kong 14-16 Oct. 1998 (Singapore: Springer-Verlag 1998), p.283-288

[10] 孙晓菲. XML 与数字图书馆. 现代图书情报技术，2000,(16): 14-15

[11] C. C. Yang. Metadata design for Chinese medicine digital library using XML. Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Maui. Hi. USA, 4-7 Jan. 2000 (Los Alamitos, CA. USA: IEEE comput.Soc.2000) , 10pp. Vol. 1.

[12] http://www.idl.pku.edu.cn

[13] http://www.calis.edu.cn/gaikuang.html

[14] 韦钰. "211 工程"是科教兴国战略的基础工程.www.gmdaly.com.cn2000 年 2 月 8 日 B1 版;

[15] http://www.calis.edu.cn/gaikuang.html

A New Method for Analyzing β-pleated Sheet

Tang Gang, Tong Genglei, Xu Jinlin and Luo Jianhua
Life Academy, Shanghai Jiao Tong University
Shanghai 200030, P.R.China
Email: l-jianhua@online.sh.cn

ABSTRACT

In this paper, we propose a new method for analyzing β-pleated sheet. In this method, we get the information of protein secondary structure from NRL_3D database at first. Then β-pleated sheet is computed its probability by dividing into the sequence of one or two or three amino acid residues. At last, we calculate the hydrophobicity, molecular mass, volume and specific volume of each sequence and analyze them. Our statistics to β-pleated sheet is the premise of predicting β-pleated sheet in protein secondary structure.

Keywords: β-pleated sheet, Statistic, Analyze.

1. INTRODUCTION

The structure of protein can be classified into four layers: primary structure, secondary structure, tertiary structure and quarternary structure. Primary structure is defined as the sequence of protein. Secondary structure is the regular arrangement of polypeptide chain. Tertiary structure is expressed as close globular conformation with particular peptide chain. Quarternary structure is defined as integrated style of each subunit of oligomeric protein in space. The normal secondary structure consists of α-helix, β-pleated sheet and β-turn. β-pleated sheet can be thought of as a particular helix, because in it there is only two amino acid residues in each helix. β-pleated sheet can be classified two types: parallel β-pleated sheet and anti-parallel β-pleated sheet. In parallel β-pleated sheet, φ is -119°, ψ is 113°, and ω is 180°. In anti-parallel β-pleated sheet, φ is –139°, ψ is 135°, and ω is 180°. Form energy, anti-parallel β-pleated sheet is more stable than parallel β-pleated sheet. In β-pleated sheet, the polypeptide chain is indented pleated conformation. -NH- and -C=O- in neighboring peptide chain come into being regular hydrogen bond.

2. MATERIAL AND METHOD

Our data is from NRL_3D database (web site: http://www-nbrf.georgetown.edu/pir/).Having queried database with selecting ID, Sequence, Feature: FtKey, we get 14791 protein records. This is the style of one protein records:

NRL3D:1TN3
>P1;1TN3
F;26-36/Region: helix (right hand alpha)
F;46-60/Region: helix (right hand alpha)
F;1-8,14-21,129-136/Region: beta sheet
F;64-68,108-113,118-122/Region: beta sheet
F;6-16/Disulfide bonds:
F;33-132/Disulfide bonds:
F;108-124/Disulfide bonds:
F;72,76,103,106,107/Site: Asp, Glu, Gly, Glu, Asn
F;99,101,106,121/Site: Gln, Asp, Glu, Asp
>P1;1TN3
ALQTVCLKGT KVHMKCFLAF TQTKTFHEAS EDC
ISRGGT LSTPQTGSEND ALYEYLRQSV GNEAEIWLGL
NDMAAEGTWV DMTGARIAYK NWETEITAQP
DGGKTENCAV LSGAANGKWF DKRCRDQLPY
ICQFGIV*

The first section is the description about secondary structure, disulfide bond and site. And the second section is the protein sequence. Because of some inaccuracy and redundancy of the database, it is necessary to process the data. (1) delete the redundant data and inaccuracy data; (2) delete all records where there is X (X represents any amino acid); (3) let D, E take the place of B, let Q, N take the place of Z, so one records is divided into two records.

Let Ω be a the statistical sample space, x be a kind of Amino Acid String (AAS) in Ω, n_x are the number of x in Ω, s be the length of AAS, and $P(x,s)$ be the probability of x in Ω. Then $P(x,s)$ is define as,

$$P(x,s)=\frac{n_x}{\sum_{x\in\Omega} n_x} \quad (1)$$

Where $\sum_{x\in\Omega} n_x$ expresses the sum of Ω's AAS.

Let Ω_β be the Ω's subspace which secondary structure all are β-pleated sheet, $n_{\beta,x}$ is the number of x kind of AAS in Ω_β, and $\mathbf{P}_\beta(x,s)$ be the probability of x in Ω_β. Then $\mathbf{P}_\beta(x,s)$ is define as,

$$\mathbf{P}_\beta(x,s)=252\frac{n_{\beta,x}}{\sum_{x\in\Omega_\beta} n_{\beta,x}} \quad (2)$$

Where $\sum_{x\in\Omega_\beta} n_{\beta,x}$ expresses the sum of Ω_β's AAS.

When s=2, there are 20^2 different permutation in twenty kinds of amino acids, or there are 400 data in all $\mathbf{P}_\beta(x,s)$. When s=3, there are 8000 data in $\mathbf{P}_\beta(x,s)$, The data is increased with the length s of AAS at 20 times.

Any amino acid sequence of protein can be changed into a data sequence of $\mathbf{P}_\beta(x,s)$.

3. RESULTS AND DISCUSSIONS

3.1 Statistic results and analysis to two amino acid residues

When the s of $\mathbf{P}_\beta(x,s)$ is equal to 2, after calculating the probability $\mathbf{P}_\beta(x,s)$, hydrophobicity, molecular mass, volume and specific volume of each sequence, we obtain 400

data points. Fig.1 reflects the relationship between probability and hydrophobicity. Fig.2 reflects the relationship between probability and specific volume.

Fig.1 and 2 show that with the hydrophobility or specific volume increasing, the probability will increase that the sequence is β-pleated sheet will increase.

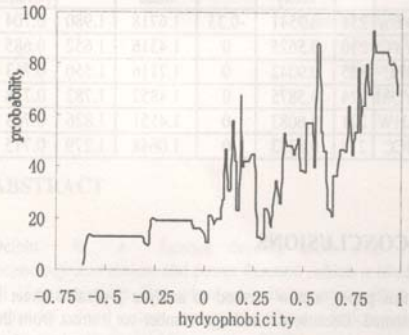


Fig.1 hydrophobicity vs probability

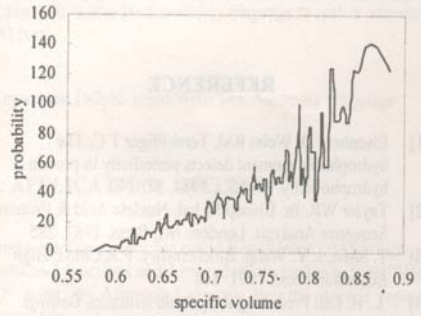


Fig.2 specific volume vs probability

Fig.3 reflects the relationship between probability and molecular mass. Fig.4 reflects the relationship between probability and volume. These two figures demonstrate that there is not definite relation between probability and molecular mass or volume. But there is a limited regularity that the probability will be large with the molecular mass being more than 1.5 or volume being more than 1.8.

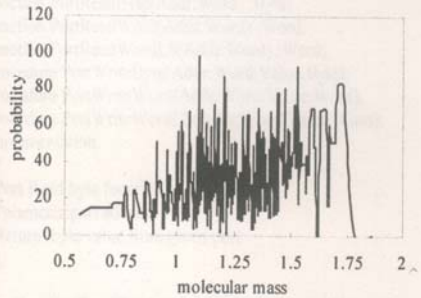


Fig.3 molecular mass vs probability

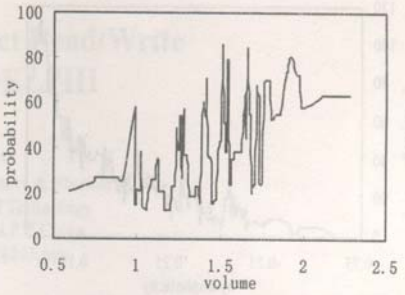


Fig.4 volume vs probability

Table.1 is the sequence with high probability and their five parameters. Table.2 is the sequence with low probability and their parameters. From there two tables, we find that the sequence with high probability has high hydrophobicity, large molecular mass, large volume and large specific volume. Vise versa, the sequence with low probability has low hydrophobicity, small molecular mass, small volume and small specific volume.

Table1 Sequence parameters with high probability

Seq.	Prob.	Hydrophobicity	Charge	Molecular mass	Volume	Specific volume
VV	136	0.8625	0	0.9914	1.400	0.847
VI	122	0.8938	0	1.0616	1.534	0.866
WV	112	0.8188	0	1.4268	1.281	0.791
IY	112	0.6875	0	1.3818	1.802	0.798
YC	111	0.6188	0	1.3563	1.511	0.672
IV	110	0.8938	0	1.0616	1.534	0.866

Table.2 Sequence parameters with low probability

Seq.	Pro.	Hydrophobicity	Charge	Molecular mass	Volume	Specific volume
PP	1	0.5125	0	0.9712	1.227	0.758
PD	1	-0.05	-0.5	1.0611	1.169	0.669
DP	2	-0.05	-0.5	1.0611	1.169	0.669
DD	2	-0.6125	-1	1.1509	1.111	0.579
PN	4	0.225	0	1.0562	1.202	0.689
PE	4	0.0125	-0.5	1.1312	1.306	0.701

3.2 Statistic results and analysis to three amino acid residues

The same as the analysis to two amino acid residues, the figure about the relationship between the sequence probability and its parameter is as follows:

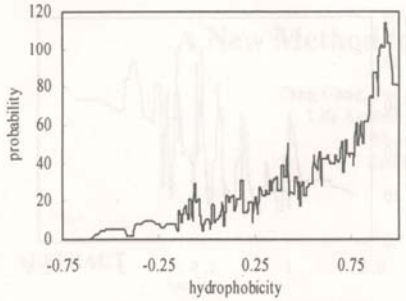


Fig.5 hydrophobicity vs probability

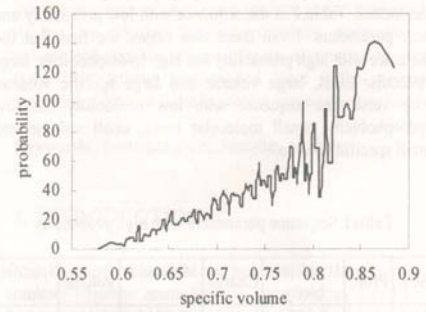


Fig.6 specific volume vs probability

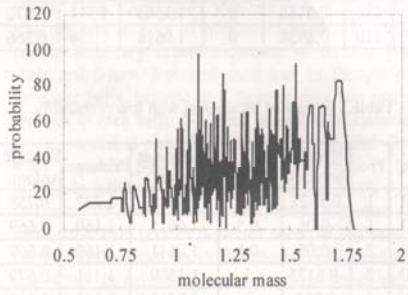


Fig.7 molecular mass vs probability

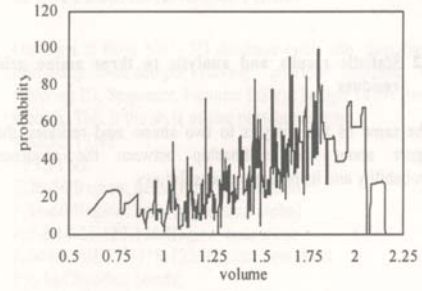


Fig.8 volume vs probability

Table.3 is the sequence with high probability and their five parameters. From the table and figures, our conclusion is the same as when the sequence length is equal to 2.

Table 3 Sequence parameters with high probability

Seq.	Pro.	Hydrophobicity	Charge	Molecular mass	Volume	Specific Volume
WEW	251	0.3541	-0.33	1.6718	1.980	0.704
YYC	230	0.5625	0	1.4316	1.652	0.685
IFC	225	0.9042	0	1.2116	1.550	0.763
WQM	224	0.5875	0	1.4852	1.782	0.718
LHW	221	0.6083	0	1.4551	1.826	0.763
ICC	218	0.8333	0	1.0648	1.279	0.715

4. CONCLUSIONS

In this paper, a new method to analyze β-pleated sheet is presented. Because of the huge number we extract from the database, our statistical results are more reliable than others are. The results are the premise of predicting β-pleated sheet and the future works will be the prediction on β-pleated sheet.

REFERENCE

[1] Eisenberg D, Weiss RM, Terw illiger T C. The hydrophobic moment detects periodicity in protein hydrophobicity. PNAS. , 1984, 81: 140

[2] Taylor WR. In: Bishop M Jed. Nucleic Acid & Protein Sequence Analysis. London: IRL Press, 1987. 285

[3] T. Shen, J. Y. Wang. Biochemistry. P.R.China: High Education Press, 1991. 148

[4] L. H. Lai. Predicting the protein structure. Beijing: Beijing University Press, 1993

A Solution for Direct Read/Write I/O Port in DELPHI

Wang Yingjun
Dept. of Engineering Structure & Mechanics
Wuhan University of Technology
Wuhan430070,Hubei,P.R.China
Email: wyj95950@263.net

ABSTRACT

Delphi is a famous develop tools with OOP technology ,convenient and power function ,which is liked by all of programmers .But, when we write program with it for industrial control system , operate the all outer devices linking computer , and control I/O port directly, this tool is a blemish in an otherwise perfect things and it doesn't support functions and commands to Read/Write I/O port as C++. This paper applied the methods of assembly language to realize operating I/O port.

Keywords: Delphi ,Read/Write I/O, Assemply Language

1. APPLICATION METHOD

First,we define an unit file called ReadWriteIO.PAS,which includes 3 functions and 3 procedures to realize to Read/Write IO ports by Assemply Language, thus the proceed speed is very fast.
Second,we add the unit file to a project .When it is in need of using the functions and procedures in other unit files,we add the ReadWriteIO.PAS to the unit files.

2. PROGRAMME BILL

```
unit Port95;
interface
function PortReadByte(Addr:Word) :Byte;
function PortReadWord(Addr:Word) :Word;
function PortReadWordLS(Addr:Word) :Word;
procedure PortWriteByte(Addr:Word;Value:Byte);
procedure PortWriteWord(Addr:Word;Value:Word);
procedure PortWriteWordLS(Addr:Word;Value:Word);
implementation
{*
*Port Read byte function
*Parameter:port address
*Return :byte value from given port
*}`

function PortReadByte(Addr:Word) :Byte;assembler;register;
asm
    MOV   DX,AX
    IN AL,DX
end;
{*
*HIGHT SPEED Port Read Word function
*Parameter:port address
*Return:word value from given port
*Comment:may problem with some cards and computers
*that can't to access whole word ,usually it works.
*}

function
PortReadWord(Addr:Word) :Word;assembler;register;
asm
    MOV   DX,AX
    IN AX,DX
end;
{*
*LOW SPEED Port Read Word function
*Parameter:port address
*Return:word value from given port
*Comment:work in cases,only to adjust DELAY if needed
*}
function        PortReadWordLS(Addr:Word)        :Word;
assembler;register;
const
   Delay=150;
   //depending of CPU speed and cards speed
asm
    MOV   DX,AX
    IN AL,DX   //read LSB port
    MOV ECX,Delay
    @1:
    LOOP   @1   //dealy between two cards
    XCHG AH,AL
    INC DX     //port+1
    IN AL,DX   //read MSB port
    XCHG AH,AL //restore bytes order
    end;

    {*Port Read byte function*}

   procedure PortWriteByte(Addr:Word;Value:Byte);
   assembler;register;asm
    XCHG   AX,DX
    OUT   DX,AL
end;
    {*
      *HIGHT SPEED Port Write Word function
      *Comment:may problem with some cards
      *and computers that can't to access whole
      * word ,usually it *works.
     *}
```

```
procedure PortWriteWord(Addr:Word;Value:Word);
assembler;register;asm
    XCHG   AX,DX
    OUT    DX,AX
end;
    {*
      * LOW SPEED Port Write Word procedure
    *}

procedure PortWriteWordLS(Addr:Word;Value:Word) :Word;
assembler;register;
const
   Delay=150;
   //depending of CPU speed and cards speed
asm
    XCHG AX,DX
    OUT DX,AL
    MOV ECX,Delay
    @1:
    LOOP   @1
    XCHG AH,AL
    INC DX
    OUT DX,AL
end;
end. //unit define end
```

3. EXAMPLE

Now, we use aboved method to operate printer(LPT1) which port address is 378H.We send a char or word to printer with aboved procedure PortWriteByte(Addr:Word;Value:Byte) and
PortWriteWord(Addr:Word;Value:Word).
For example:

```
Var
    Addr:word;
    Value1:Byte;
    Value2:word;
    Addr:=$378;
    Value1:='a';
    Value2:=$8000;
PortWriteByte(Addr,Value1) ;
PortWriteWord(Addr,Value2);
```

The printing result is true .

REFERENCES

[1] 季雪岗,王晓辉等,Delphi编程疑难详解,北京,人民邮电出版社,2000年7月

[2] 蒋方帅,Delphi5程序员指南,北京,人民邮电出版社,2000年8月

[3] 沈美明,温冬婵,IBM—PC汇编语言程序设计,北京,清华大学出版社,1991年6月

[4] Donna N.Tabler: IBM PC Assembly Language John Wiley & Sons,Inc.,1985

Multi -fractal Algorithm

Dan Liu,Yuanhui Li,Yue Ma,Yicheng Jin
The Institute of Nautical Science & Technology, Dalian Maritime University
Dalian, Liaoning 116026, P.R.China
Email: dliu_dlmu@263.net

ABSTRACT

A multi-fractal algorithm is proposed, This algorithm is an effective method to simulate phenomena with multi-degree, such as percolation simulation, trees emulation and fractals generation .the frame chart is also given.

Keywords: Fractal, Multi-Fractal, Extended Multi-Fractal

1. INTRODUCTION

Multi-fractal is brought forward for studying nonuniformity and anisotropic phenomena in the nature; it is different from simply fractal in that it is related to degree character and direction.

So single dimension cannot describe its full characters, which must be expressed by the measure of multi-fractal or the sequence chart of dimension. Multi-fractal is often connected with randomicity. The definition as follows is used by most now.

Definition1: Suppose X is a subset of d-dimension Euclidean space, apply proper recursive or iterative graduation, and endow unchangeable measure μ . α is a parameter related to graduation, note the subset of X in the n-step $X_n(\alpha)$, if $X_\alpha = \lim_{n\to\infty} X_n(\alpha)$ is a fractal set, then it is called the fractal subset of (X,μ). If in the graduation, the fractal subset generated by (X,μ) can be expressed by the combination of some fractal subset, and each fractal subset has different fractal dimension, so call the fractal set is a multi-fractal.

It is difficult to describe multi-fractal with computer, In order to simplify the description; matrix substitutions are useful to propose the concept of extended multi-fractal, which includes some special kinds of multi-fractal.

Definition2: Consider substitutions from $\{0,1\}$ to the set of $N \times N$ matrices for some positive integer N . Let

$$\sigma(0) = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{pmatrix}$$

$\sigma(1)$ will be random, taking values in a set of $N \times N$ 0-1 valued matrices $\{U_1, U_2, \cdots, U_m\}$; specifically, there are positive numbers $p_1, p_2 \cdots, p_m$ with $\sum_{i=1}^{m} p_i = 1$, $P\{\sigma(1) = U_i\} = p_i (1 \le i \le m)$. Then $A^n = \sigma^n(1)$ is a $N^n \times N^n$ matrix, let

$$I_k^n = \left[(k-1)N^{-n}, kN^{-n}\right], 1 \le k \le N^n$$

$$I_{kl}^n = I_k^n \times I_l^n$$

Consider set

$$F_n = \bigcup_{k,l} \{I_{kl}^n : A_{kl}^n = 1\}$$

Then F_n decrease to a compact subset of the unit square

$$F = \lim_{n\to\infty} F_n$$

call F is an extended multi-fractal.

2. MULTI-FRACTAL ALGORITHM

Given an initiatory data matrix $P^{(0)}$, which is a $m \times n$ 0-1 valued matrix, $P^{(0)} = \left(p_{ij}^{(0)}\right)_{m\times n}$, $p_{ij}^{(0)} \in \{0,1\}, i = 1,2,\cdots,m, j = 1,2,\cdots,n$. On the plane, choose a square $[0,1]\times[0,1]$, divide it into $m \times n$ rectangles which length of sides are $\frac{1}{n}, \frac{1}{m}$, $p_{ij}^{(0)}$ is situated in the $m \times n$ rectangle gridding cells. The coordinates of the vertexes on the cross of the rectangle gridding cell where the $p_{ij}^{(0)}$ is situated are $\left(\frac{j-1}{n}, \frac{m-i+1}{m}\right)$ and $\left(\frac{j}{n}, \frac{m-i}{m}\right)$. Now, apply the iterative operation to matrix $P^{(0)} = \left(p_{ij}^{(0)}\right)_{m\times n}$.

The first step, divide every rectangles which lengths of sides are $\frac{1}{n}, \frac{1}{m}$ into $m \times n$ smaller rectangles, apply once iterative operation to $P^{(0)}$, obtain one $m^2 \times n^2$ probability measure matrix $P^{(1)} = \left(p_{kl}^{(1)}\right)_{m^2\times n^2}$, which is the Kronecker product of $P^{(0)}$ and itself, namely,

$P^{(1)} = \left(p_{kl}^{(1)}\right)_{m^2 \times n^2} = P^{(0)} \otimes P^{(0)}$, Where, $\otimes$ expresses the Kronecker product of the matrix, the coordinate of the vertexes on the cross of the matrix where the $p_{kl}^{(1)}$ is situated are $\left(\frac{l-1}{n^2}, \frac{m^2-k+1}{m^2}\right)$ and $\left(\frac{l}{n^2}, \frac{m^2-k}{m^2}\right)$.

Repeat the process

The N-th step, one $m^{N+1} \times n^{N+1}$ probability measure matrix $P^{(N)} = \left(p_{st}^{(N)}\right)_{m^{N+1} \times n^{N+1}}$ will be obtained, which is the Kronecker product of $P^{(0)}$ and $P^{(N-1)}$ or the N times Kronecker product of $P^{(0)}$ and itself. Namely,

$$P^{(N)} = \left(p_{st}^{(N)}\right)_{m^{N+1} \times n^{N+1}} = \underbrace{P^{(0)} \otimes P^{(0)} \otimes \cdots \otimes P^{(0)}}_{N} = P^{(0)} \otimes P^{(N-1)}$$

The elements of $P^{(N)}$ can be wrote as $P_{st}^{(N)} = P_{ab} P_{uv}^{(N-1)}$,where, $a = 1,2,\cdots,m, b = 1,2,\cdots,n; u = 1,2,\cdots,m^N$, $v = 1,2,\cdots,n^N$; $s = 1,2,\cdots,m^{N+1}$, $t = 1,2,\cdots,n^{N+1}$.

Accordingly, unit square was divided into $m^{N+1} \times n^{N+1}$ rectangles which lengths of sides is $\frac{1}{n^{N+1}}, \frac{1}{m^{N+1}}$, the coordinate of vertexes on the cross of the matrix where the $P_{st}^{(N)}$ is situated are $\left(\frac{t-1}{n^{N+1}}, \frac{m^{N+1}-s+1}{m^{N+1}}\right)$ and $\left(\frac{t}{n^{N+1}}, \frac{m^{N+1}-s}{m^{N+1}}\right)$.

When the demand of calculation precision is achieved, if $P_{st}^{(N)}$=1, then fill the （s,t）-th rectangle, else this rectangle gridding cell would be empty. In this way, one distribution of multi –fractal on the plane can be obtained. In a general way,

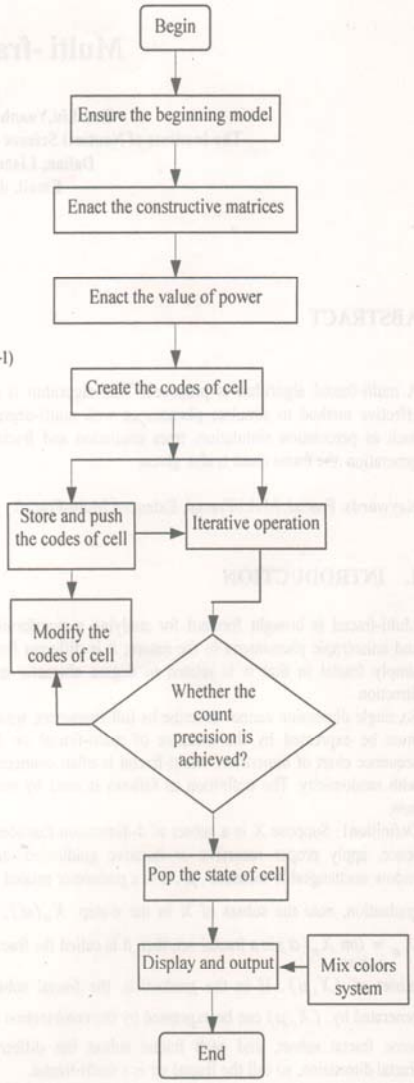
(1) if m=n, the result is the self-similar set which has only one degree

(2) if m≠n, the result is the self-affined set which has two degrees.

For simulating the fractal percolation phenomena, the detail of the algorithm is as follows:

Get one unit square on the plane, divide it into 3×3 rectangles which lengths of sides is $\frac{1}{3}$,According to the iterative rule of the fractal percolation base set, given a 3×3 0-1 valued initiatory data matrix, construct the table of matrices and the value of power probability, substitute every 0 with $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$; Substitute every 1 with the constructional matrix chosen from the constructional table of matrices by power probability, until the demand of calculation precision is achieved.

3. THE FRAME CHART OF THE ALGORITHM



REFERENCES

[1] Chayes J.T.,Chayes L. and Durrett R. Connectivity properties of Mandelbrot's percolation process,Probab. Th. Rel. Fields77,307-324(1988).

[2] Falconer K.Projections of random Cantor sets, Journal of Theoretical Prob.2, 65-70(1989).

[3] Izmailov R., PokpovskII A., Vladimirov A., Visualization of polynomials, Comput. & Graphics 1,95-105 (1996).