2002 International Symposium on
Distributed Computing and Applications to
Business, Engineering and Science

DCABES 2002

PROCEEDINGS

Editor in Chief Xu Wenbo
Associate Editor Guo Qingping

Wuhan University of Technology Press
Wuhan, China

2002 International Symposium on

Distributed Computing and Applications to

Business, Engineering and Science

DCABES 2002

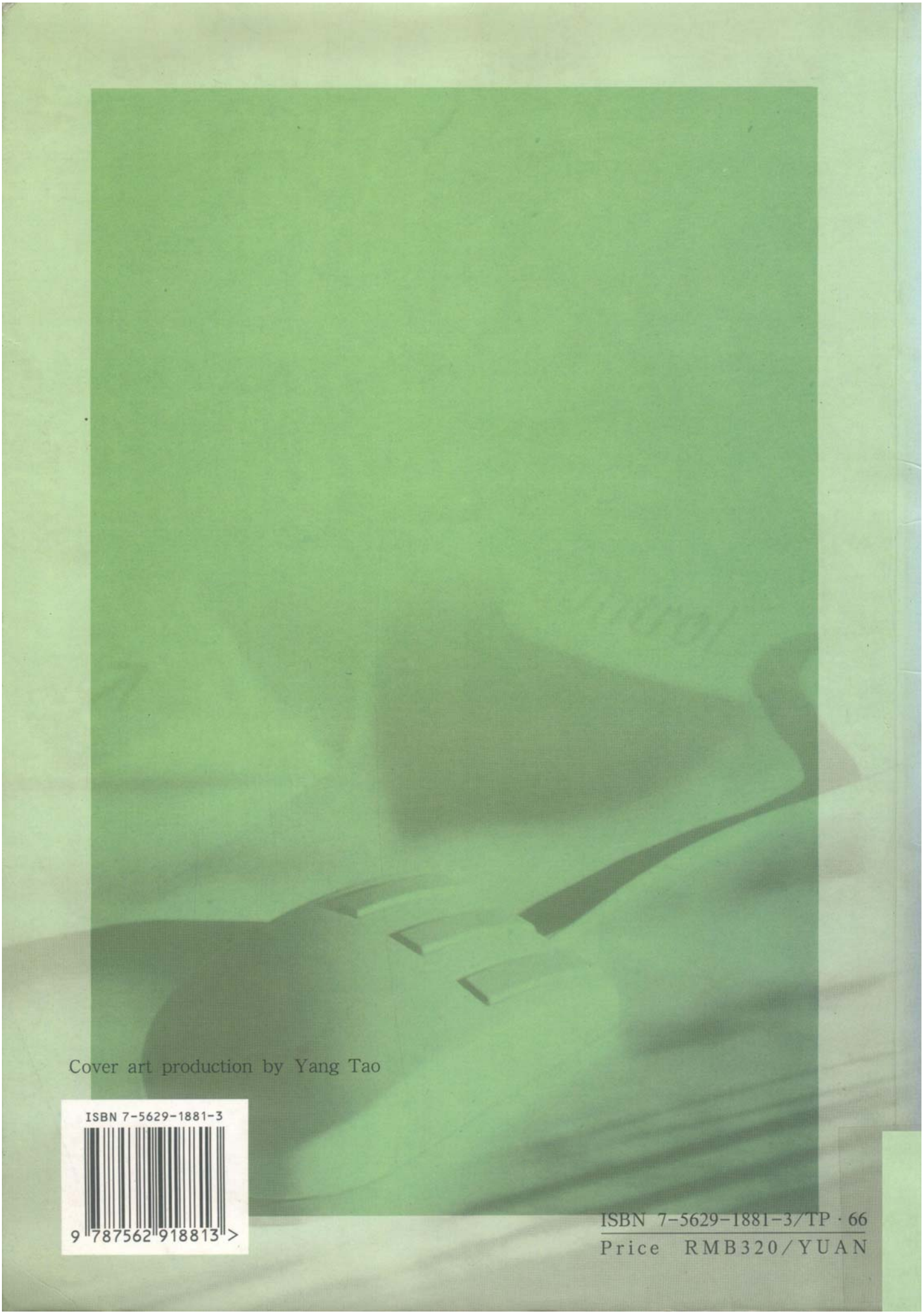
PROCEEDINGS

Editor in Chief Xu Wenbo

Associate Editor Guo Qingping

Wuhan University of Technology Press

Wuhan,China



Cover art production by Yang Tao

ISBN 7-5629-1881-3

9 787562 918813 >

ISBN 7-5629-1881-3/TP · 66

Price RMB320/YUAN

图书在版编目（CIP）数据
2002 年 电 子 商 务 、 工 程 暨 科 学 领 域 分 布 式 计 算 和 应 用 论 文 集 DCABES2002 PROCEEDINGS / 须文波主编.－武汉：武汉理工大学出版社，2002.12
ISBN 7-5629-1881-3

Ⅰ.2 … Ⅱ．须 … 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（2002）第 096896 号

Organized by
SYU Southern Yangtze University
Co-organized by
WUT Wuhan University of Technology
Sponsored by
MOE Ministry of Education, China
NSFC National Nature Science Foundation of China
AVAYA Corporation

DCABES 2002 PROCEEDINGS © Editor in Chief Xu Wenbo

Editorial production by Tian Daoquan and Duan Chao Cover art production by Yang Tao
Published by ***Wuhan University of Technology Press,Wuhan,China*** Tel:+86-(0)27-87394107
Address: 122 Luoshi Road,Wuhan,China Post Code: 430070
Printed in Wuhan,China by Wuhan University of Technology Printery Post Code: 430070
880mm×1230mm sixteenmo 29 sheets
2200 k Characters
First Edittion December 2002 First Impression December 2002
ISBN 7-5629-1881-3 Price RMB 320.00
QoS of Book Printing and Bounding is guaranteed by the printery.

Parallel & Distributed Algorithms

Distributed system and Distributed Computing

Grid Computing and Parallel Processing

Network and Web Application

Database and Engineering Applications

Preface

High-performance computing is increasingly being used in all aspects of modern society. It is well known that the distributed parallel computing plays an important role in the HPC. In recent years, more and more attentions have been put on to the distributed parallel computing.

It was my pleasure that the DCABES2002 conference had received a great number of papers submitted cover a wide range of topics, such as Parallel/Distributed Algorithms, Distributed System and Distributed Computing, Grid Computing and Parallel Processing, Network and Applications, Database and Engineering Applications.

Papers submitting to the conference come from over 16 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 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, Prof. Qingping Guo, 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 Professor David Keyes, Professor H.M. Liddell, Dr Anne Trefethen, Prof. Nengchao Wang, Prof. Chi, Xue-Bin as well as Prof. Shi, Zhongzhi, 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) and Southern Yangtze University. It should also be mentioned that the AVAYA (www. avaya.com.cn) made some contribution to the conference.

Finally I should also thank Professor Yaolin Gu,Yaming Bo and Mrs Li Liu ,for their efforts in conference organizing activities, my postgraduate students, such as Ms Jing Liu, Mr Lu Huang,Yonggang Hu,Zhiyang Yao ,Ms Ling Bao,Mr Wei Fang,Kezhong Lu, for their time and help. Without their time and efforts this conference cannot be organized smoothly.

Enjoy your stay in Wuxi. Hope to meet you again at the DCABES 2003.

Professor Wenbo Xu,
Chair of the DCABES2002
Dean of School of Information Technology
Southern Yangtze University
Jiangsu, China

Chair of Scientific Committee

Professor Xu, Wenbo, Southern Yangtze University

Co-Chair of Scientific Committee

Professor Guo, Qingping, Wuhan University of Technology
Dr. Lai, Choi-Hong. ,University of Greenwich, U.K.

Chair of Organizing Committee

Professor Xu, Wenbo, Southern Yangtze University

Scientific Committee

Professor Xu, Wenbo, Southern Yangtze University, China
Professor Guo, Qingping, Wuhan University of Technology, China
Dr. Lai, Choi-Hong ,University of Greenwich, U.K.
Professor Jesshope, C., University of Hull, U.K.
Professor Kang, L.S., Wuhan University, China
Dr. Lee, John, Hong Kong Polytechnic University, China
Professor Liddell, H. M, Queen Mary, University of London, U.K.
Dr. Lin, H.X., Delft University of Technology, the Netherlands
Professor Lin, P., National University of Singapore, Singapore
Dr. Loo, Alfred, Hong Kong Lingnan University, China
Dr. Ng, Michael, University of Hong Kong, China
Professor Sun, J., Academia Sinica, China
Professor Gu ,Yaolin, Southern Yangtze University, China
Mr. Tsui, Thomas, Chinese University of Hong Kong, China
Professor Zhang, J., University of Kentucky, U.S.A.

Local Organizing Committee

Professor Xu, Wen-bo, Southern Yangtze University, China
Professor Chen, Jian, Southern Yangtze University, China
Professor Ji, Zeng-rui, Southern Yangtze Computer Research Institute, China
Professor Xu, Bao-guo, Southern Yangtze University, China
Dr. Bo, Ya-ming, Southern Yangtze University, China
Professor Wang, Shi-tong, Southern Yangtze University, China
Professor Leng, Wen-hao, Ship Scientific Research Institute, China
Professor Zhang, Ji-wen, Southern Yangtze University, China
Professor Xu, Zhen-yuan, Southern Yangtze University, China
Professor Feng, Bin, Southern Yangtze University, China
Professor Gu, Yao-lin, Southern Yangtze University, China

Editorial Board

Algorithms for Terascale Computation of PDEs and PDE-constrained Optimization

David E. Keyes
Mathematics & Statistics Department, Old Dominion University
Norfolk, VA 23529-0077 USA
E-mail: dkeyes@odu.edu

ABSTRACT

For the simulation of systems governed by partial differential equations (PDEs) at the terascale we introduce Jacobian-free Newton-Krylov-Schwarz methods and their relatives. A key concept is the exploitation of multiple discrete representations of the underlying continuous operator, to converge fully to a representation of high fidelity through a series of inexpensive and stable steps based on representations of lower fidelity. Advances in object-oriented software engineering have enabled the construction of internally complex software systems in which these algorithmic elements can be combined modularly, recursively, and relatively efficiently in parallel, while presenting a programming environment that allows the user to function at a rather high level.

Keywords: terascale simulation, partial differential equations, optimal algorithms, Newton-Krylov-Schwarz methods

1. INTRODUCTION

Computational peak performance on full-scale scientific applications, as tracked by the Gordon Bell prize, has increased by four orders of magnitude since the prize was first awarded in 1988 — twenty-five times faster than can accounted for by Moore's Law alone. The extra factor comes from process concurrency, which is as much as 8,192-fold on the $100M "ASCI White" machine at Lawrence Livermore, currently ranked as the world's second most powerful, after the new Japanese "Earth Simulator". The latter was recently clocked at more than 35 trillion floating point operations per second (Top/s) on the LINPACK benchmark and at 26.6 Top/s on a climate application [27]. Though architectural concurrency is easy to achieve, algorithmic concurrency to match is less so in scientific codes. Intuitively, this is due to global domains of influence in many problems presented to the computer as implicitly discretized operator equations — implicitness being all but legislated for the multiscale systems of global climate, transonic airliners, petroleum reservoirs, tokamaks, etc., the simulation of which justifies expenditures for the highest-end machines.

A key requirement of candidate solution algorithms is mathematical optimality. This means a convergence rate as independent as possible of discretization parameters. In practice, linear systems require a hierarchical, multilevel approach to obtain rapid linear convergence. Nonlinear systems require a Newton-like approach to obtain asymptotically quadratic convergence. The concept of optimality can also be extended into the physical modeling regime to include continuation schemes and physics-informed preconditioning, so that multiple scale problems are attacked with a manageable number of scales visible to the numerics at any given stage.

In this context, optimal parallel algorithms for PDE simulations of Jacobian-free Newton-Krylov type, preconditioned with Schwarz and Schur domain decompositions, including multilevel generalizations of Schwarz, are coming into prominence. For large systems with strong nonlinearities robustification techniques have been developed, including pseudo-transient continuation, parameter continuation, grid sequencing, model sequencing, and non-linear preconditioning. These improvements in nonlinear rootfinding have made it possible for large-scale PDE-constrained optimization problems (e.g., design, control, parameter identification — usually the ultimate problems behind the proximate PDEs) to be placed into the same domain decomposed algorithmic framework as the PDE, itself.

The architecture of the terascale systems available until recently, built around hierarchical distributed memory, appears hostile to conventional sequential optimal PDE algorithms, but is ultimately suitable apart from reservations about limited memory bandwidth. The distributed aspects must be overcome with judicious combinations of message-passing and/or shared memory program models. The hierarchical aspects must be overcome with register blocking, cache blocking, and prefetching. Algorithms for these PDE-based simulations must be highly concurrent, straightforward to load balance, latency tolerant, cache friendly (with strong temporal and spatial locality of reference), and highly scalable (in the sense of convergence rate) as problem size and processor number are increased in proportion. The goal for algorithmic scalability is to fill up the memory of arbitrarily large machines while preserving constant (or at most logarithmically growing) running times with respect to a proportionally smaller problem on one processor. Domain-decomposed multilevel methods are natural for all of these *desiderata*. Domain decomposition is also natural for the software engineering of simulation codes: valuable extent code designed for a sequential PDE analysis can often be "componentized" and made part of an effective domain-decomposed, operator-split preconditioner.

For a pair of web-downloadable full-scale reviews documenting these themes more fully, see [17, 18]. This page-limited chapter skims these reviews at a high level, emphasizing the importance of domain decomposition to large-scale scientific computing.

2. THE NEWTON-KRYLOV-SCHWARZ FAMILY OF ALGORITHMS

Many problems in engineering and applied physics can be written in the form

$$V \frac{\partial u}{\partial t} + F(u) = 0 \qquad (1)$$

where u is a vector of functions depending upon spatial variables x and t, F is a vector of spatial differential operators acting on u, and V is a diagonal scaling matrix with nonnegative diagonal entries. If all of the equations are "prognostic" then V has strictly positive diagonal entries; but we may also accommodate the case of entirely steady-state equations, V=0, or some combination of positive and zero diagonal entries, corresponding to prognostic equations for

some variables and steady-state constraints for others. Steady-state equations often arise from a *priori* equilibrium assumptions designed to suppress timescales faster than those of dynamical interest, e.g., acoustic waves in aerodynamics, gravity waves in geophysics, Alfv é n waves in magnetohydrodynamics, etc.

Semidiscretizing in space to approximate *F(u)* with *f (u)*, and in time with implicit Euler, we get the algebraic system:

$$(\frac{V}{\tau^l})u^l + f(u^l) = (\frac{V}{\tau^l})u^{l-1} \quad (2)$$

Higher-order temporal schemes are easily put into this framework with the incorporation of additional history vectors with appropriate weights on the right-hand side. We are not directly concerned with discretization or the adaptivity of the discretization to the solution in this chap- ter. However, the achievement of nonlinear consistency by Newton's method on each time step is motivated by a desire to go to higher order than the pervasive standard of no better than first-order in time and second-order in space. Because *f* may be highly nonlinear, even a steady-state numerical analysis is often made to follow a pseudo-transient continuation until the ball of convergence for Newton's method for the steady-state problem is reached. In this case, time accuracy is not an issue, and τ^l becomes a parameter to be placed at the service of the algorithm [16].

Whether discretized time accurately or not, we are left at each time step with a system of nonlinear algebraic equations (2), written abstractly as $F^l(u^l)=0$. We solve these systems in sequence for each set of discretized spatial gridfunctions, u^l, using an inexact Newton method. The resulting linear systems for the Newton corrections involving the large but sparse Jacobian of F^l with respect to instantaneous or lagged iterates $u^{l,k}$, are solved with a Krylov method, relying only on Jacobian-vector multiplications. (Here, $u^{l,0} \equiv u^{l-1}$, and $u^{l,k} \rightarrow u^l$, as $k \rightarrow \infty$ in a Newton iteration loop on inner index *k*.) Due to Jacobian ill-conditioning, The Krylov method needs to be preconditioned for acceptable inner iteration convergence rates, and the preconditioning is the "make-or-break" aspect of an implicit code. The other phases possess high concurrency and parallelize well already, if properly load balanced, being made up of vector updates, inner products, and sparse matrix-vector products.

The job of the preconditioner is to approximate the action of the Jacobian inverse in a way that does not make it the dominant consumer of memory or cycles in the overall algorithm and (most importantly) does not introduce idleness through chained data dependencies, as in Gaussian elimination. The true inverse of the Jacobian is usually dense, reecting the global Green's function of the continuous linearized PDE operator it approximates, and it is not obvious that a good preconditioner approximating this inverse action can avoid extensive global communication. A good preconditioner saves time and space by permitting fewer iterations in the Krylov loop and smaller storage for the Krylov subspace than would be required in its absence. An additive Schwarz preconditioner accomplishes this in a localized manner, with an approximate solve in each subdomain of a partitioning of the global PDE domain. Applying any subdomain preconditioner within an additive Schwarz framework tends to increases floating point rates over the same preconditioner applied globally, since the smaller subdomain blocks maintain better cache residency. Combining a Schwarz preconditioner with a Krylov iteration method inside an inexact Newton method leads to a synergistic parallelizable nonlinear boundary value problem solver with a classical name: Newton-Krylov-Schwarz (NKS). In the remainder of this section, we build up NKS from the outside inwards.

Inexact Newton Methods

We use the term "inexact Newton method" to denote any nonlinear iterative method for solving $F(u)=0$ through a sequence $u^k = u^{k-1} + \lambda^k \delta u^k$, where δu^k approximately satisfies the true Newton correction equation

$$F'\left(u^{k-1}\right)\delta u^k = -F\left(u^{k-1}\right) \quad (3)$$

in the sense that the linear residual norm $\| F'(u^{k-1}) \delta u^k + F(u^{k-1}) \|$ is sufficiently small. Typically the right-hand side of the linear Newton correction equation, which is the nonlinear residual $F(u^{k-1})$, is evaluated to full precision, so the inexactness arises from incomplete convergence employing the true Jacobian, freshly evaluated at u^{k-1}, or from the employment of an inexact Jacobian for $F'(u^{k-1})$.

Newton-Krylov Methods

A Newton-Krylov (NK) method uses a Krylov method, such as GMRES [26], to solve Eq. (3) for δu^l. From a computational point of view, one of the most important characteristics of a Krylov method for the linear system $Ax=b$ is that information about the matrix A needs to be accessed only in the form of matrix-vector products in a relatively small number of carefully chosen directions. When the matrix A represents the Jacobian of a discretized system of PDEs, each of these matrix-vector products is similar in computational and communication cost to a stencil update phase (or "global flux balance") of an explicit method applied to the same set of discrete conservation equations, or to a single finest-grid "work unit" in a multigrid method. NK methods are suited for nonlinear problems in which it is unreasonable to compute or store a true full Jacobian, where the action of A can be approximated by discrete directional derivatives.

Newton-Krylov-Schwarz Methods

A Newton-Krylov-Schwarz (NKS) method combines a Newton-Krylov method, such as Newton-GMRES [6], with a Krylov-Schwarz (KS) method, such as restricted additive Schwarz [9]. If the Jacobian *A* is ill-conditioned, the Krylov method will require an unacceptably large number of iterations. In order to control the number of Krylov iterations, while obtaining concurrency proportional to the number of processors, they are preconditioned with domain-decomposed additive Schwarz methods [28]. The system is transformed into the equivalent form $B^{-1}Ax=B^{-1}b$ through the action of a preconditioner, *B*, whose inverse action approximates that of *A*, but at smaller cost. It is in the choice of preconditioning that the battle for low computational cost and scalable parallelism is usually won or lost. In KS methods, the preconditioning is introduced on a subdomain-by-subdomain basis through a conveniently computable approximation to a local Jacobian. Such Schwarz-type preconditioning provides good data locality for parallel implementations over a range of parallel granularities, allowing significant architectural adaptability.

Schwarz Methods

Schwarz methods [7, 11, 28, 32] create concurrency at a desired granularity algorithmically and explicitly through partitioning, without the necessity of any code dependence analysis or special compiler. Generically, in continuous or discrete settings, Schwarz partitions a solution space into n subspaces, possibly overlapping, whose union is the original space, and forms an approximate inverse of the operator in each subspace. Algebraically, to solve the discrete linear system, $Ax = f$, let Boolean rectangular matrix R_i extract the i^{th}

subset of the elements of x defining an algebraic subspace: $x_i = R_i x$, and let $A_i \equiv R_i A R_i^T$ be invertible within the i^{th} subspace. Then the additive Schwarzapproximate inverse is defined as

$$B_{ASM}^{-1} = \sum_i R_i^T A_i^{-1} R_i \quad (4)$$

From the PDE perspective, subspace decomposition is domain decomposition. B^{-1} is formed out of (approximate) local solves on (possibly overlapping) subdomains.

In the grid-based context of a PDE, Boolean operators R_i and $R_i^T, i = 1,...,n$, represent gather and scatter (communication) operations, mapping between a global vector and its i^{th} subdomain support. When A derives from an elliptic operator and R_i is the characteristic function of unknowns in a subdomain, optimal convergence (independent of dim(x) and the number of partitions) can be proved, with the addition of a coarse grid, which is denoted with subscript "0": $B_{ASM}^{-1} = R_0^T A_0^{-1} R_0 + \sum_{i>0} R_i^T A_i^{-1} R_i$. Here, R_0 is a conventional geometrically based multilevel interpolation operator. It is an important freedom in practical implementations that the coarse grid space need not be related to the fine grid space or to the subdomain partitioning. The $A_i^{-1}(i > 0)$ in B^{-1} are often replaced with inexact solves in practice, such as a multigrid V-cycle. The exact forward matrix-vector action of A in $B^{-1}A$ is still required, even if inexact solves are employed in the preconditioner.

Table 1 Theoretical condition number estimates $\kappa(B^{-1}A)$, for self-adjoint positive-definite elliptic problems [28] and corresponding iteration count estimates for Krylov-Schwarz based on an idealized isotropic partitioning of the domain in three dimensions.

Preconditioning	$\kappa(B^{-1}A)$	*Iter.*
Point Jacobi	$O(h^{-2})$	$O(N^{1/3})$
Domain Jacobi	$O(hH)^{-1}$	$O((N/P)^{1/6})$
1-level Additive Schwarz	$O(h^{-2})$	$O(P^{1/3})$
2-level Additive Schwarz	$O(1)$	$O(1)$

Condition number estimates for $B^{-1}A$ are given in the first column of Table 1 in terms of the quasi-uniform mesh parameter h, and subdomain parameter H. The two-level Schwarz method with generous overlap has a condition number that is independent of the fineness of the discretization and the granularity of the decomposition, which implies perfect algorithmic scalability. However, there is an increasing implementation overhead in the coarse-grid solution required in the two-level method that offsets this perfect algorithmic scalability. In practice, a one-level method is often used, since it is amenable to a perfectly scalable implementation. Alternatively, a two-level method is used but the coarse level is solved only approximately, in a trade-off that depends upon the application and the architecture. These condition number results are extensible to nonself-adjointness, mild indefiniteness, and inexact subdomain solvers. The theory requires a "sufficiently fine" coarse mesh, H, for the first two of these extensions, but computational experience shows that the theory is often pessimistic.

The restricted additive Schwarz Method (RASM) eliminates interprocess communication during the interpolation phase of the additive Schwarz technique [9]. In particular, if we decompose a problem into a set of overlapping subdomains i, the conventional additive Schwarz method is a three-phase process consisting of first collecting data from neighboring subdomains via global-to-local restriction operators R^i, then performing a local linear solve on each subdomain A_i^{-1}, and finally sending partial solutions to neighboring subdomains via the local-to-global prolongation operators R_i^T . The RASM preconditioner performs a complete restriction operation but does not use any communication during the interpolation phase, denoted instead as $R_i'^T$. This provides the obvious benefit of a 50% reduction in nearest-neighbor communication overhead. In addition, experimentally, it preconditions better than the original additive Schwarz method over a broad class of problems [9], for reasons that are beginning to be understood [8].

Although the spectral radius, $\rho(I-B^{-1}A)$, may exceed unity, the spectrum, $\sigma(I-B^{-1}A)$, is profoundly clustered, so Krylov acceleration methods work well on the preconditioned solution of $B^{-1}Ax = B^{-1}f$. Krylov-Schwarz methods typically converge in a number of iterations that scales as the square-root of the condition number of the Schwarz-preconditioned system. For convergence scalability estimates, assume one subdomain per processor in a d-dimensional isotropic problem, where $N = h^{-d}$ and $P = H^{-d}$. Then iteration counts may be estimated as in the last column of Table 1.

The proof of these estimates is generally approached via an algebra of projection operators, $P_i \equiv R_i^T A_i^{-1} R_i A$. The ratio of upper bound to lower bound of the spectrum of the sum of the orthogonal projections Pi is an estimate of the condition number for $B^{-1}A = \sum_i P_i$. Since $\| P_i \| \leqslant 1$, the upper bound follows easily from the geometry of the decomposition and is a generally a constant related to the number of colors required to color the subdomains. The lower bound depends crucially upon the partitioning of the solution space. Without a coarse subspace to support the solution at subdomain boundaries, the fine space contributions must fall rapidly to zero from finite values in the subdomain interiors, resulting in high H_1 "energy" inversely proportional to the overlap distance over which the solutions must decay.

For simple intuition behind this table consider the following: errors propagate from the interior to the boundary in steps that are proportional to the largest implicit aggregate in the preconditioner, whether pointwise (in N) or subdomainwise (in P). The use of overlap in going from Domain Jacobi to Additive Schwarz avoids the introduction of high energy at near discontinuities at sub-domain boundaries. The two-level method projects out low-wavenumber errors rapidly at the price of solving a global problem.

Only the two-level method scales perfectly in convergence rate (constant, independent of N and P), like a traditional multilevel iterative method [4, 5, 14, 30]. However, the two-level method shares with multilevel methods a nonscalable cost-per-iteration from the necessity of solving a coarse-grid system of size $O(P)$. Unlike recursive multilevel methods, a two-level Schwarz method may have a rather fine coarse grid, for example, $H = O(h^{1/2})$, which potentially makes it less scalable overall. Parallelizing the coarse grid solve is necessary. Neither extreme of a fully distributed or a fully redundant coarse solve is optimal, but rather something in between. When reuse is possible, storing a parallel inverse can be cost-effiective [31].

When it appears additively in the Schwarz precondiioner, the coarse grid injects some work that potentially spoils the "single-program, multiple data" (SPMD) parallel programming paradigm, in which each processor runs an identical image over local data. For instance, the SPMD model would not hold if one subset of processors worked on the

coarse grid problem concurrently to the others each working on subdomains. Therefore, in two-level SPMD implementations, other Schwarz preconditioner polynomials than the purely additive are used in practice. A preconditioner may be defined that solves the fine subdomains concurrently in the standard way, and then assembles a new residual and solves the coarse grid in a separate phase. This leads to the method denoted "Hybrid II" in [28]:

$$B^{-1} = A_0^{-1} + (I - A_0^{-1}A)(\sum_{i=1}^{n} A_i^{-1}) \cdot$$

The subspace inverses are typically done approximately, as in the purely additive case.

Readers uncomfortable with the appearance of the Schwarz formula $A^{-1} \approx \sum_i R_i^T A_i^{-1} R_i$, implying that the inverse of the sum is well approximated by the sum of the inverses in subspaces, may benefit from recalling an exact result from eigenanalysis. Let $\{r_i\}_i^N = 1$ be a complete set of orthonormal row (left) eigenvectors for an SPD matrix A. Then $r_iA = a_ir_i$, or $a_i = r_i A r_i^T$, for corresponding eigenvalues a_i. Then, we have the representations of A and A^{-1} as sums over subspaces,

$$A = \sum_{i=1}^{N} r_i^T a_i r_i, A^{-1} = \sum_{i=1}^{N} r_i^T a_i^{-1} r_i = \sum_{i=1}^{N} r_i^T (r_i A r_i^T)^{-1} r_i$$

The latter is nothing but a special case of the Schwarz formula! In practice, invariant subspaces are far too expensive to obtain for practical use in Schwarz, and their basis vectors are general globally dense, resulting in too much storage and communication in forming restrictions and prolongations. Characteristic subspaces of subdomains, in contrast, provide locality and sparsity, but are not invariant upon multiplication by A, since the stencils overlap subdomain boundaries. Choosing good decompositions is a balance between conditioning and parallel complexity, in practice.

Contrasting Decomposition Methods

It is worthwhile to emphasize the architectural advantages of Schwarz-type domain decomposition methods *vis-à-vis* other mathematically useful decompositions.

Given the operator equation $Lu = f$ in Ω, and a desire for either concurrent or sequential "divide-and-conquer", one can devise operator decompositions $L = \sum_j L_j$, function-space decompositions $u = \sum_j u_j \phi_j$, or domain decompositions $\Omega = U_j \Omega_j$.

Let us contrast an example of each on the parabolic PDE in two space dimensions

$$\frac{\partial u}{\partial t} + [L_x + L_y]u = f(x,y,t) \; in \; \Omega \qquad (5)$$

with $u = 0$ on $\partial\Omega$, where

$$L_x \equiv -\frac{\partial}{\partial x} a_x(x,y) \frac{\partial}{\partial x} + b_x(x,y) \frac{\partial}{\partial x}, (a_x > 0)$$

and with a corresponding form for Ly. Upon implicit time discretization

$$[\frac{I}{\Delta t} + L_x + L_y]u^{(l+1)} = [\frac{I}{\Delta t}]u^{(l)} + f \equiv \tilde{f}$$

we get an elliptic problem at each time step.

The Alternating Direction Implicit (ADI) method is an example of operator decomposition. Proceeding in half- steps, one each devoted to the x- and y-directions, we write

$$[\frac{I}{\Delta t/2} + L_x]u^{(l+1/2)} = [\frac{I}{\Delta t/2} - L_y]u^{(l)} + f$$

$$[\frac{I}{\Delta t/2} + L_y]u^{(l+1)} = [\frac{I}{\Delta t/2} - L_x]u^{(l+1/2)} + f$$

The overall iteration matrix mapping $u^{(l)}$ to $u^{(l+1)}$ is factored into four sequential substeps per time step: two sparse matrix-vector multiplies and two sets of unidirectional bandsolves. If the data is alternately laid out in unidirectional slabs on the processors, so as to allow each set of unidirectional bandsolves to be executed independently, then we have perfect parallelism within substeps, but, global data exchanges *between* substeps. In other words, computation and communication each scale with the bulk size of the data of the problem.

A Fourier or spectral method is an example of a function-space decomposition. We expand

$$u(x,y,t) = \sum_{j=1}^{N} a_j(t)\phi_j(x,y) \cdot$$

Enforcing Galerkin conditions on Eg. (5) with the basis functions ϕ_i, we get

$$\frac{d}{dt}(\phi_i, u) = (\phi_i, L_u) + (\phi_i, f), i = 1,...N.$$

Plugging the expansion into the Galerkin form,

$$\sum_{j=1}^{N} (\phi_i, \phi_j) \frac{da_j}{dt} = \sum_{j=1}^{N} (\phi_i, L\phi_j)a_j + (\phi_j, f), i = 1,...,N.$$

Inverting the mass matrix, $M \equiv [(\phi_j, \phi_i)]$ on both sides, and denoting the stifiness matrix by $K \equiv [(\phi_j, L\phi_i)]$, we get a set of ordinary differential equations for the expansion coefficients:

$$\dot{a} = M^{-1}Ka + M^{-1}g \cdot$$

If the basis functions are orthogonal and diagonalize the operator, then M and K are diagonal, and these equations perfectly decouple, creating N-fold concurrency for the evolution of the spectral components. However, in applications, it is necessary to frequently reconstitute the physical variable u. This is true for interpreting or visualizing the model and also for handling possible additional terms of the PDE in physical space in a "pseudo-spectral" approach, since it is unlikely that practically arising operators readily lead to orthogonal eigenfunctions for which there are fast transforms. Transforming back and forth from physical to spectral space on each iteration leads, again, to an algorithm where the computation and the communication together scale with the problem size, and there is all-to-all communication.

An additive Schwarz domain decomposition method for this problem has been described already. We replace $Au = f$ by $B_{ASM}^{-1} Au = B_{ASM}^{-1} f$ and solve by a Krylov method. There are several Krylov steps per time step, each requiring a matrix-vector multiplies with $B_{ASM}^{-1} A$. Due to the concurrency implied by the sum, there is parallelism on each subregion. However the dominant communication is nearest-neighbor data exchange, whose size scales as the perimeter (resp., surface in three dimensions), compared to the computation, whose size scales as the area (resp., volume). Therefore, domain decomposition possesses excellent scalability properties with respect to implementation on distributed memory computers. There is a need for a small global sparse linear system solve in some problems, to obtain mathematical optimality. (This is not necessary for the parabolic problem considered above.) Though this small problem requires global communication (either to set up redundant instances, solved concurrently, or to carry out a collaborative solution) and demands analysis and extreme care to keep subdominant, it escapes the bulk communication

burdens of the other approaches.

Physics-based Preconditioning

An important class of preconditioners for the Jacobian-free Newton-Krylov method, complementary to the domain-split parallelism of Schwarz, is physics-based operator splitting. The operator notation for the right-preconditioned, matrix-free form of the method is:

$$J(u) = B_{split}^{-1} v \approx \frac{F(u + \varepsilon B_{split}^{-1} v) - F(u)}{\varepsilon}, \qquad (6)$$

where "split" denotes a preconditioning process handled in an operator-split manner. Many operator-split time integration methods have been developed based on insight from the physics of the underlying system. It is well understood that operator-split methods have limitations as solvers, thus they most likely also have limitations as preconditioners. However, they still provide an interesting class of preconditioners for the Jacobian-free Newton-Krylov method

The essential insight of physics-based preconditioning is that preconditioner in a Newton-Krylov method maps a nonlinear residual to an approximate state-vector correction, namely, the Newton update. Such a map implicitly resides in most interative procedures of computational physics. The use of operator-split solvers as preconditioners for Jacobian-free Newton-Krylov appears not to have a long history, but is rapidly developing. See instances for time-independent reaction diffusion equations [24], time-dependent MHD equations [10], steady state incompressible Navier-Stokes equations [19, 25], and time-dependent incompressible Navier-Stokes equations [20, 25]. Also in [20], a standard approximate linearization method used for phase-change heat conduction problems, has been employed as a preconditioner for a JFNK solution of phase-change heat conduction problems.

3. PARALLEL IMPLEMENTATION OF NKS USING PETSc

To implement NKS methods on distributed memory parallel computers, we employ the "Portable, Extensible Toolkit for Scientific Computing" (PETSc) [1, 2], a library that attempts to handle through a uniform interface, in a highly efficient way, the low-level details of the distributed memory hierarchy. Examples of such details include striking the right balance between buffering messages and minimizing buffer copies, overlapping communication and computation, organizing node code for strong cache locality, preallocating memory in sizable chunks rather than incrementally, and separating tasks into one-time and every-time subtasks using the inspector/executor paradigm. The benefits to be gained from these and from other numerically neutral but architecturally sensitive techniques are so significant that it is efficient in both the programmer-time and execution-time senses to express them in general purpose code. Among other important packages implementing Newton-Krylov in parallel, we mention Aztec [15], KINSOL [29], NITSOL [23], and the Iterative Template Library (ITL) [21]

PETSc is a large and versatile package integrating distributed vectors, distributed matrices in several sparse storage formats, Krylov subspace methods, preconditioners, and Newton-like nonlinear methods with built-in trust region or linesearch strategies and continuation for robustness. It has been designed to provide the numerical infrastructure for application codes involving the implicit numerical solution of PDEs, and it sits atop MPI for portability to most parallel machines. The PETSc library is written in C, but may be accessed from user codes written in C, FORTRAN, and C++. PETSc version 2, first released in June 1995, has been downloaded thousands of times by users worldwide. PETSc has many features relevant to PDE analysis, including matrix-free Krylov methods, blocked forms of parallel preconditioners, and various types of time-stepping.

When well tuned, large-scale PDE codes spend almost all of their time in two phases: flux computations to evaluate conservation law residuals, where one aims to have such codes spent almost *all* their time, and sparse linear algebraic kernels, which are a fact of life in implicit methods. Altogether, four basic groups of tasks can be identified based on the criteria of arithmetic concurrency, communication patterns, and the ratio of operation complexity to data size within the task. These four distinct phases, present in most implicit codes, are vertex-based loops, edge-based loops, recurrences, and global reductions. Each of these groups of tasks has a distinct proportion of work to datasize to communication requirements and each stresses a different subsystem of contemporary high-performance computers. In the language of a vertex-centered code, in which the data is stored at cell vertices, these tasks are as follows:

- Vertex-based loops
  - —state vector and auxiliary vector updates
- Edge-based "stencil op" loops
  - —residual evaluation, Jacobian evaluation
  - — Jacobian-vector product (often replaced with matrix-free form, involving residual evaluation)
  - —interpolation between grid levels
- Sparse, narrow-band recurrences
  - — (approximate) factorization, back substitution, relaxation/smoothing
- vector inner products and norms
  - —orthogonalization/conjugation
  - —convergence progress checks and stability heuristics

Vertex-based loops are characterized by work closely proportional to datasize, pointwise concurrency, and no communication.

Edge-based "stencil op" loops have a large ratio of work to datasize, since each vertex is used in many discrete stencil operations, and each degree of freedom at a point (momenta, energy, density, species concentration) generally interacts with all others in the conservation laws — through constitutive and state relationships or directly. There is concurrency at the level of the number of edges between vertices (or, at worst, the number of edges of a given "color" when write consistency needs to be protected through mesh coloring). There is local communication between processors sharing ownership of the vertices in a stencil.

Sparse, narrow-band recurrences involve work closely proportional to data size, the matrix being the largest data object and each of its elements typically being used once. Concurrency is at the level of the number of fronts in the recurrence, which may vary with the level of exactness of the recurrence. In a preconditioned iterative method, the recurrences are typically broken to deliver a prescribed process concurrency; only the quality of the preconditioning is thereby affected, not the final result. Depending upon whether one uses a pure decomposed Schwarz-type preconditioner, a truncated incomplete solve, or an exact solve, there may be no, local only, or global communication in this task.

Vector inner products and norms involve work closely proportional to data size, mostly pointwise concurrency, and global communication.

Based on these characteristics, one anticipates that

vertex-based loops, recurrences, and inner products will be *memory bandwidth-limited*, whereas edge-based loops are likely to be only *load/store-limited*. However, edge-based loops are vulnerable to *internode bandwidth* if the latter does not scale. Inner products are vulnerable to *internode latency* and *network diameter*. Recurrences can resemble some combination of edge-based loops and inner products in their communication characteristics if preconditioning fancier than simple Schwarz is employed. For instance, if incomplete factorization is employed globally or a coarse grid is used in a multilevel preconditioner, global recurrences ensue.

Analysis of a parallel aerodynamics code reimplemented in PETSc [13] shows that, after tuning, as expected, the linear algebraic kernels run at close to the aggregate memory bandwidth limit on performance, the flux computations are bounded either by memory bandwidth or instruction scheduling (depending upon the ratio of load/store units to floating-point units in the CPU), and parallel efficiency is bounded primarily by slight load imbalances at synchronization points.

4. TERASCALE OPTIMAL PDE SIMULATIONS (TOPS)

Under the Scientific Discovery through Advanced Computing initiative of the U.S. Department of Energy (http://www.science.doe.gov/scidac/), a nine-institution team is building an integrated software infrastructure center (ISIC) that focuses on developing, implementing, and supporting optimal or near optimal schemes for PDE simulations and closely related tasks, including optimization of PDE-constrained systems, eigenanalysis, and adaptive time integration, as well as implicit linear and nonlinear solvers. The Terascale Optimal PDE Simulations (TOPS) Center is researching and developing and will deploy a toolkit of open source solvers for the nonlinear partial differential equations that arise in many application areas, including fusion, accelerator design, global climate change, and the collapse of supernovae. These algorithms —primarily multilevel methods —aim to reduce computational bottlenecks by one or more orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible.

Along with usability, robustness, and algorithmic efficiency, an important goal of this ISIC is to attain the highest possible computational performance in its implementations by accommodating to the memory bandwidth limitations of hierarchical memory architectures.

PDE simulation codes require implicit solvers for multiscale, multiphase, multiphysics phenomena from hydrodynamics, electromagnetism, radiation transport, chemical kinetics, and quantum chemistry. Problem sizes are typically now in the millions of unknowns; and with emerging large-scale computing systems and inexpensive clusters, we expect this size to increase by a factor of a thousand over the next five years. Moreover, these simulations are increasingly used for design optimization, parameter identification, and process control applications that require many repeated, related simulations.

The TOPS ISIC is concerned with five PDE simulation capabilities: adaptive time integrators for stiff systems, nonlinear implicit solvers, optimization, linear solvers, and eigenanalysis. The relationship between these areas is depicted in Figure 1. In addition, TOPS emphasizes two cross-cutting topics: software integration (or interoperability) and high-performance coding techniques for PDE applications.

Optimal (and nearly optimal) complexity numerical algorithms almost invariably depend upon a hierarchy of approximations to "bootstrap" to the required highly accurate final solution. Generally, an underlying continuum (infinite dimensional) high fidelity mathematical model of the physics is discretized to "high" order on a "fine" mesh to define the top level of the hierarchy of approximations. The representations of the problem at lower levels of the hierarchy may employ other models (possibly of lower physical fidelity), coarser meshes, lower order discretization schemes, inexact linearizations, and even lower floating-point precisions. The philosophy that underlies our algorithmics and software is the same as that of this chapter —to make the majority of progress towards the highly resolved result through possibly low-resolution stages that run well on high-end distributed hierarchical memory computers.

The ingredients for constructing methods based on hierarchies of approximations are remarkably similar, be it for solving linear systems, nonlinear problems, eigenvalue problems, or optimization problems, namely:

1. A method for generating several discrete problems at different resolutions (for example on several grids),
2. An inexpensive (requiring few floating point operations, loads, and stores per degree of freedom) method for iteratively improving an approximate solution at a particular resolution,
3. A means of interpolating (discrete) functions at a particular resolution to the next finer resolution,
4. A means of transferring (discrete) functions at a particular resolution to the next coarser resolution (often obtained trivially from interpolation).

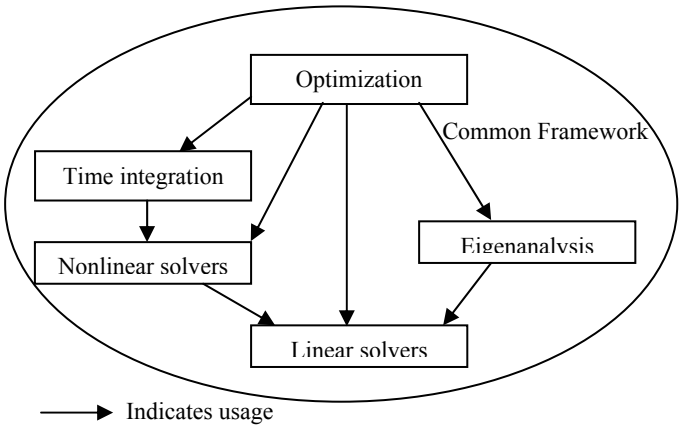


Figure 1 An arrow from A to B indicates that A typically uses B. Optimization of systems governed by PDEs requires repeated access to a PDE solver. The PDE system may be steady state or time-dependent. Time-dependent PDEs are typically solved with implicit temporal differencing. After choice of the time-integration scheme, they, in turn, require the same types of nonlinear solvers that are used to solve steady-state PDEs. Many algorithms for nonlinear problems of high dimension generate a sequence of linear problems, so linear solver capability is at the core. Eigenanalysis arises inside of or independently of optimization. Like direct PDE analysis, eigenanalysis generally depends upon solving a sequence of linear problems. All of these five classes of problems, in a PDE context, share grid-based data structures and considerable parallel software infrastructure. Therefore, it is compelling to undertake them together.

Software should reflect the simplicity and uniformity of these ingredients over the five problem classes and over a wide range of applications. With experience we expect to achieve a

reduction in the number of lines of code that need to be written and maintained, because the same code can be reused in many circumstances.

The efforts defined for TOPS, the co-PIs joining to undertake them, and the alliances proposed with other groups have been chosen to exploit the present opportunity to revolutionize large-scale solver infrastructure, and lift the capabilities of dozens of DOE's computational science groups as an outcome. The co-PIs' current software (e.g., Hypre [12], PETSc [1], ScaLAPACK [3], SuperLU [22]), though not algorithmically optimal in many cases, and not yet as interoperable as required, is in the hands of thousands of users, and has created a valuable experience base. Just as we expect the user community to drive research and development, we expect to significantly impact the scientific priorities of users by emphasizing optimization (inverse problems, optimal control, optimal design) and eigenanalysis as part of the solver toolkit.

Optimization subject to PDE-constraints is a particularly active subfield of optimization because the traditional means of handling constraints in black-box optimization codes — with a call to a PDE solver in the inner loop —is too expensive. We are emphasizing "simultaneous analysis and design" methods in which the cost of doing the optimization is a small multiple of doing the simulation and the simulation data structures are actually part of the optimization data structures.

Likewise, we expect that a convenient software path from PDE analysis to eigenanalysis will impact the scientific approach of users with complex applications. For instance, a PDE analysis can be pipelined into the scientific added-value tasks of stability analysis for small perturbations about a solution and reduced dimension representations (model reduction), with reuse of distributed data structures and solver components.

The motivation behind TOPS is that most PDE simulation is ultimately a part of some larger scientific process that can be hosted by the same data structures and carried out with many of the same optimized kernels as the simulation, itself. We intend to make the connection to such processes explicit and inviting to users, and this will be a prime metric of our success. The organization of the effort ows directly from this program of "holistic simulation": Terascale software for PDEs should extend from the analysis to the scientifically important auxiliary processes of sensitivity analysis, modal analysis and the ultimate "prize" of optimization subject to conservation laws embodied by the PDE system.

5. CONCLUSIONS

The emergence of the nonlinearly implicit Jacobian-free Newton-Krylov-Schwarz family of methods has provided a pathway towards terascale simulation of PDE-based systems. Domain decomposition is desirable for possessing a communication cost that is subdominant to computation — even optimal order computation, linear in the problem size — and fixed in ratio, as problem size and processor count are scaled in proportion.

Large-scale implicit computations have matured to a point of practical use on distributed/shared memory architectures for static-grid problems. More sophisticated algorithms, including solution adaptivity, inherit the same features *within* static-grid phases, of course, but require extensive additional infrastructure for dynamic parallel adaptivity, rebalancing, and maintenance of efficient, consistent distributed data structures.

While mathematical theory has been crucial in the development of NKS methods, their most successful application also depends upon a more-than-superficial understanding of the underlying architecture and of the physics being modeled. In the future, as we head towards petascale simulation and greater integration of complex physics codes in full system analysis and optimization, we expect that this interdisciplinary interdependence will only increase.

6. ACKNOWLEGMENTS

The author thanks Xiao-Chuan Cai, Omar Ghattas, Bill Gropp, Dana Knoll, and Barry Smith for long-term collaborations on parallel algorithms, and Satish Balay, Paul Hovland, Dinesh Kaushik, and Lois McInnes from the PETSc team at Argonne National Laboratory (along with Gropp and Smith and others) for their wizardry in implementation.

7. REFERENCES

[1] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, Efficient management of parallelism in object-oriented numerical software libraries, Modern Software Tools in Scientific Computing, Birkhauser, 1997, pp. 163-201.

[2] Users' guide to the Portable, Extensible Toolkit for Scientific Computing, version 2.3.1, http://www.mcs.anl.gov/petsc/, 2002.

[3] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, ScaLAPACK users' guide, SIAM, 1997.

[4] A. Brandt, Multi-level adaptive solutions to boundary value problems, Math. Comp. 31 (1977), 333{390.

[5] Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics, Tech. report, von Karman Institute, 1984.

[6] P. N. Brown and Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, SIAM J. Sci. Stat. Comput. 11 (1990), 450-481.

[7] X.-C. Cai, Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations, Technical Report 461, Courant Institute, 1989.

[8] X.-C. Cai, M. Dryja, and M. Sarkis, RASHO: A restricted additive Schwarz preconditioner with harmonic overlap, Proceedings of the 13th International Conference on Domain Decomposition Methods, Domain Decomposition Press, 2002.

[9] X.-C. Cai and M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (1999), 792-797.

[10] L. Chacon, D. A. Knoll, and J. M. Finn, An implicit nonlinear resistive and Hall MHD solver, J. Comput. Phys. 178 (2002), 15-36.

[11] M. Dryja and O. B. Widlund, An additive variant of the Schwarz alternating method for the case of many subregions, Tech. Report 339, Department of Computer Science, Courant Institute, 1987.

[12] R. D. Falgout and U. M. Yang, Hypre: a library of high performance preconditioners, Lecture Notes in Computer Science, vol. 2331, Springer-Verlag, 2002, pp. 632-641.

[13] W. D. Gropp, D. K. Kaushik D. E. Keyes, and B. F. Smith, High performance parallel implicit CFD, Parallel Computing 27 (2001), 337-362.

[14] W. Hackbusch, Iterative methods for large sparse linear systems, Springer, 1993.

[15] S. A. Hutchinson, J. N. Shadid, and R. S. Tuminaro,

Aztec user's guide: Version 1.1, Tech. Report SAND95-1559, Sandia National Laboratories, October 1995.
[16] C. T. Kelley and D. E. Keyes, Convergence analysis of pseudo-transient continuation, SIAM J. Numer. Anal. 35 (1998), 508-523.
[17] D. E. Keyes, Terascale implicit methods for partial differential equations, Recent Advances in Numerical Methods for Partial Differential Equations and Applications, Contemporary Mathematics vol. 306, AMS, 2002, pp. 29-84.
[18] D. A. Knoll and D. E. Keyes, Jacobian-free Newton-Krylov methods: A survey of approaches and applications, submitted to J. Comput. Phys., 2002.
[19] D. A. Knoll and V.A. Mousseau, On Newton-Krylov multigrid methods for the incompressible NavierStokes equations, J. Comput. Phys. 163 (2000), 262-267.
[20] D. A. Knoll, W. B. VanderHeyden, V. A. Mousseau, and D. B. Kothe, On preconditioning Newton-Krylov methods in solidifying ow applications, SIAM J. Sci. Comput. 23 (2001), 381-397.
[21] L.-Q. Lee and A. Lumsdaine, The iterative template library, submitted to ACM Transactions on Mathematical Software, 2002.
[22] X. S. Li and J. W. Demmel, SuperLU DIST: A scal able distributed-memory sparse direct solver for unsymmetric linear systems, submitted to ACM Transactions on Mathematical Software; also available as Lawrence Berkeley National Laboratory tech report LBNL-49388, 2002.
[23] Homer F. Walker Michael Pernice, NITSOL: A Newton iterative solver for nonlinear systems, SIAM J. Sci. Stat. Comput. 19 (1998), 302-318.
[24] V.A. Mousseau, D. A. Knoll, and W.J. Rider, Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion, J. Comput. Phys. 160 (2000), 743-765.
[25] M. Pernice and M. D. Tocci, A multigrid preconditioned Newton-Krylov method for the incompressible Navier-Stokes equations, SIAM J. Sci. Comput. 23 (2001), 398-418.
[26] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986), 856-869.
[27] S. Shingu, H. Takahara, H. Fuchigami, M. Yamada, Y. Tsuda, W. Ohfuchi, Y. Sasaki, K. Kobayashi, T. Hagiwara, S.-I Habata, M. Yokokawa, H. Itoh, and K. Otsuka, A 26.58 Top/s global atmospheric simulation with the spectral transform method on the earth simulator, Proceedings of SC2002, to appear, 2002.
[28] B. F. Smith, P. Bjfirstad, and W. D. Gropp, Domain decomposition: Parallel multilevel methods for elliptic partial differential equations, Cambridge University Press, 1996.
[29] A. G. Taylor and A. C. Hindmarsh, User documentation for KINSOL: a nonlinear solver for sequential and parallel computers, Tech. Report UCRL-ID131185, Lawrence Livermore National Laboratory,July 1998.
[30] U. Trottenberg, A. Schuller, and C. Oosterlee, Multigrid, Academic Press, 2000.
[31] H. M. Tufo and P.F. Fischer, Fast parallel direct solvers for coarse grid problems, J. Par. Dist. Comput. 61 (2001), 151-177.
[32] J. Xu, Iterative methods by space decomposition and subspace correction, SIAM Review 34 (1991), 581-613.

A PPGAs-based Algorithm for Task Scheduling on Distributed Multiprocessor Systems

Wenbo Xu Jun Sun Lijuan Zhu
School of Information Technology, Southern Yangtze University,
Wuxi, Jiangsu 214036, P.R. China
E-mail: sunjun21c@163.com

ABSTRACT

The scheduling problem of arbitrary task graphs to distributed multiprocessors is generally NP-complete even with simplifying assumptions. Genetic algorithms (GAs) have been widely reckoned as a useful vehicle for combinatorial optimization problems including scheduling problem. Having studied the suggested GAs for scheduling, in this paper we propose a pseudo-parallel genetic scheduling algorithm (PPGAS) with different mutation operator from other GAs-based scheduling algorithms.

Key words: DAG, genetic algorithm, chromosome, selection, crossover, mutation.

1. INTRODUCTION

Task scheduling is one of the key elements in any distributed multiprocessor system, and an efficient algorithm can help reduce the interprocessor communication time. In static scheduling, given a parallel program modeled by a directed acyclic weight task graph, the objective of a scheduling algorithm is to minimize the overall execution time of the program by properly assigning the nodes of the graph to the processors. Even with simplifying assumptions, the multiprocessor scheduling problem has been proven to be an NP-Complete problem except for some simple cases for which an optimal solution can be obtained in polynomial time. Suboptimal solutions to the static scheduling problem can be obtained by heuristic methods. These heuristics, however, have restricted applicability in a practical environment because they have a number of fundamental problems including high time complexity, lack of scalability, and no performance guarantee with respect to optimal solutions.

Genetic algorithms (GAs) have been reckoned as an efficient vehicle for obtaining high quality or even optimal solutions for NP-complete combinatorial optimization problems. Having studied a few suggested GAs for scheduling, in this paper we propose a pseudo-parallel genetic scheduling (PPGS) algorithm in which a novel mutation operator is exploited. The rest part of the paper is organized as follows:

In Section 2, we describe the basic terminology and assumptions used in scheduling algorithms. In Section 3, we introduce the methodology of GAs and the main ideology of PPGAS. In Section 4, we formulate in detail the designation of the algorithm. In Section 5, we describe our experimental study and its results. Finally we provide some concluding remarks in the last section.

2. PROBLEM DEFINITION AND ASSUMPTIONS

A *directed acyclic weighted task graph* (DAG) is defined by a tuple $G=(V,E,C,W)$ where $V=\{n_j, j=1:v\}$ is the set of task nodes and $v=|V|$ is the number of nodes, E is the set of communication edges and $e=|E|$ is the number of edges, C is the set of edge communication costs, and W is the set of node computation costs. The value $c(n_i,n_j)\in C$ the communication cost incurred along the edge $e_{i,j}=(n_i,n_j)\in E$, which is zero if both nodes are assigned on the same processor. The value $w(n_i)\in W$ is the execution time of the node $n_i\in V$.

The *communication-to-computation-ratio* (CCR) of a parallel program is defined as its average communication cost divided by its average computation cost on a given system. It is commonly assumed that the processing elements (PEs) in the target system are connected by an interconnection network based on a certain topology in which a message is transmitted with the same speed on all links, and thus, a multiprocessor network can be represented by an undirected graph. A fully-connected network, a hypercube or mesh is a paradigm of this model.

Other relevant definitions and denotations are described as follows: DAT denotes data available time of a node on a particular PE; $ST(n_i)$ and $FT(n_i)$ denote the start-time and finish-time of n_i respectively, if node n_i is scheduled; After all nodes have been scheduled, the schedule length is defined as $\max_i\{FT(n_i)\}$ across all nodes; The *b-level* of a node is the length (sum of the computation and communication costs) of the longest path from this node to an exit node. The *t-level* of a node is the length of the longest path from an entry node to this node (excluding the cost of this node); ALAP is as late as possible start-time of a node. The objective of scheduling is to minimize the schedule length by proper allocation of the nodes to the processors and arrangement of execution sequencing of the nodes without violating the precedence constraints of a DAG, which dictate that a node cannot start execution before it gathers all of the messages from its parent nodes.

3. OVERVIEW OF GENETIC SEARCH ALGORITHMS

3.1 SIMPLE GENETIC ALGORITHMS

Genetic algorithms (GAs), a type of adaptive global-optimization search algorithms simulating the genetic and evolutionary process in natural environment, were first introduced by Holland in the 1970s. A GA is usually employed to determine the optimal solution of a specific objective function. Having abstracted the common characteristics from the genetic algorithms, Goldberg suggested a so-called Simple Genetic Algorithms (SGA), in which only three genetic operators—selection, crossover, and mutation—are applied to a population of chromosomes to improve the quality of the chromosomes which is evaluated by the fitness function of a chromosome. SGA, the embryo and foundation of other GAs, not only provides the underlying framework of a variety of GAs, but has application value itself as well. It can be defined

as

$$SGA = (C,\ E,\ P_0,\ M,\ \Phi,\ \Gamma,\ \Psi,\ T),$$

where C is encoding method of individuals, E is Fitness Function by which the chromosome of an individual is evaluated, P_0 is the Generation of the Initial Population, Φ, Γ, Ψ denote the selection, crossover and mutation operators respectively, and T is the ending conditions.

3.2 Parallel Genetic Algorithms

In order to raise the speed of the GAs and improve its performance, parallelism is exploited and many parallel genetic algorithms are developed under the environment of parallel computers or LANs. By and large, there are two types of parallel genetic algorithms: Standard Parallel Approach (SPA) and Decomposition Parallel Approach (DPA). The latter approach is used more frequently than the former one in that, to implement the SPA, a united population and a global memory are needed, and even a unified controlling mechanism must be employed to coordinate the processes of the genetic evolution and the communications among the populations.

The Decomposition Parallel Approach of parallelizing a GA is to divide the population into several partitions. The DPA-based parallel genetic algorithms can be defined as

$$PAG = (DMM,\ X,\ Z,\ \Delta,\ \Theta,\ SGA),$$

where DMM is the set of physical processing elements (PPEs) of the parallel computer or LAN on which the parallel GA is executed, X is the set of the object exchanged among PPEs, Z is the content of information exchange, Δ is the frequency of information exchange, Θ is the substitution operator when the information exchange occurs, and SGA is the Simple Genetic Algorithms executed on each PPE.

3.3 The main ideology of Pseudo-Parallel Genetic Algorithms

One striking problems of Simple Genetic Algorithms is its liability to premature, which exerts unfavorable influences on the efficacy of GAs. In order to overcome the shortcomings, by using the methodology of the parallel genetic algorithms to SGA, we can divide the global population into several partitions each of which evolve independently and among which messages are exchange at appropriate time, and thus, the diversity of the population is maintained. The algorithm is executed on a single-CPU computer and called pseudo-parallel genetic algorithm. The formulation of the PPGA is provided in the following section.

4. THE PROPOSED PSEUDO-PARALLEL GENETIC ALGORITHM FOR SCHEDULING

4.1 A Genetic Formulation of the Scheduling Problem

1). *Encoding.* We encode a valid scheduling list as a chromosome. A valid scheduling list is one in which the nodes of the DAG are in a topological order.

2). *Fitness Function.* The Fitness Function defines fitness value $F = \left(\sum w(n_i) - SL\right) \Big/ \sum w(n_i)$, where the schedule length SL is determined by using the start-time minimization method [1][2][4]. The fitness of a chromosome is therefore always bounded between 0 and 1.

3). *Generation of the Initial Population.* An initial population is generated from a set of scheduling lists which are constructed by ALAP ordering, b-level ordering, t-level ordering, sl ordering, and a random topological ordering, etc. A whole population is then generated from these ordering by performing random valid swapping of nodes in the lists.

4.2 Genetic Operator

1). *Selection Operator.* Selection operator, whose purpose is to improve the global convergence and the efficacy, is based on the evaluation of the individual chromosome. We adopt Elitist Model by which the fittest chromosome of the present population is retained and escape crossover and mutation operation, but the chromosome that has the least fitness value is eliminated.

2). *Crossover Operator.* The genetic algorithms are distinguished from other evolutionary algorithms in that the crossover operators are ushered into the algorithms. The crossover operator is critical for GAs and is the major approach to generate novel chromosomes. We consider a single-point order crossover operator defined in [4]. This order crossover operator is easy to implement and permit fast processing. The most important merit is that it never violates the precedence constraints.

3). *Mutation Operator.* As auxiliary approach to generate novel chromosomes, the mutation operator is indispensable for GAs because it determines the GAs' local search ability. In all the existing literatures on genetic algorithms of the scheduling problem, swap operations are used as mutation. By swapping some nodes, a valid topological order can be transformed into another topological order. The fatal weakness of the swap operation is its liability to generate invalid chromosomes. In this paper, we propose an insertion operation as mutation operator. It is described as follows.

(1). Pick up each individual chromosome of the population by probability p_m and do the following process.

(2). Pick up a node n_i randomly from the chromosome.

(3). Pick up another node n_j randomly from the chromosome.

(4). Processing depth-first traversal of the DAG.

(5). If n_i and n_j are neighboring in the chromosome list, execute step (6); or else, skip to step (8).

(6). If n_i is behind n_j, continue; or else, execute step (7);

If every node that is between n_i and n_j in the list is not on the same path with n_i, insert n_i into the place next behind n_j, and thus the mutation is over; or else, the mutation fails.

(7). If n_i is before n_j, then, if every node that is between n_i and n_j in the list is not on the same path with n_i, insert n_i into the place before n_j, and thus the mutation is over; or else the mutation fails.

(8). If n_i and n_j are close neighboring, then if n_i is not on the same path with n_j, swap n_i and n_j, and thus the mutation is over; or else the mutation fails.

4.3 Control Parameters

To sustain the diversity of the global population more effectively, we use adaptive control parameters as suggested by Srinivas *et al.* The adaptive crossover rate p_c is defined as

$$p_c = \frac{k_c\left(F_{\max} - F'\right)}{\left(F_{\max} - F_{avg}\right)},$$

where $F_{\max}$ is the maximum fitness value in the local population, F_{avg} is the average fitness value, F' is the fitness value of the fitter parent for the crossover, and k_c is a positive real constant less than 1.

The adaptive mutation rate p_m is defined as

$$p_m = \frac{k_m (F_{\max} - F)}{(F_{\max} - F_{avg})},$$

where F is the fitness value of the chromosome to be mutated and k_m is a positive real constant less than 1.

Two other control parameters that are critical to the performance of a GA are the population M and the number of generation T. As Kwok and Ahmad do [5], we set $M = k_p v$ and $T = k_g v$, where k_p and k_g are real constants.

4.4 The Model of Information Exchange Among Subpopulations

For efficiency we use a synchronous connected island model to achieve the information exchange among the subpopulations. In the island model, there is more than one individual chromosome within each subpopulation. The period of information exchange is set to t number of generations, which follows and exponentially decreasing sequence. The rationale is that at the beginning of the search, the diversity of the global population is high. At such early stages, exploration is more important than exploitation; therefore, each subpopulation should evolve independently for a longer period of time. When the search reaches the later stages, it is likely that the global population converges to a number of different fittest chromosomes. Thus, exploitation of more promising chromosomes is needed to void unnecessary work on optimizing the locally best chromosomes of the subpopulation that may have smaller fitness value than the global best chromosomes.

4.5 The Formulation of Pseudo-parallel Genetic Scheduling Algorithm

With the above design consideration, the pseudo-parallel genetic scheduling algorithm is outline below.

Pseudo-parallel Genetic Scheduling Algorithm:

(1). Initialize the genetic generation counter t=0;

(2). By perturbing predefined topological orderings of the DAG (e.g., ALAP ordering, b-*level* ordering, etc.), generate an initial global population $P(t)$ with the size equal to M. According to the synchronous connected island model, divide $P(t)$ into subpopulations:

$$P(t) = \{P_1(t), P_2(t), \cdots, P_i(t), \cdots P_n(t)\},$$

where *n* is the number of the subpopulations.

(3). Compute the fitness value of each individual chromosome in $P_i(t) \quad (i = 1,2,\cdots,n)$ respectively.

(4). For each subpopulation $P_i(t) \quad (i = 1,2,\cdots,n)$, execute selection, crossover and mutation operation respectively:

$$P_i'(t) \leftarrow selection \quad [P_i(t)]$$

$$P_i''(t) \leftarrow crossover \quad [P_i'(t)]$$

$$P_i'''(t) \leftarrow mutation \quad [P_i''(t)]$$

(5). Compute the fitness value of each individual chromosome in $P_i(t) \quad (i = 1,2,\cdots,n)$ respectively.

(6). According to the synchronous connected island model, exchange information among $P_i(t) \quad (i = 1,2,\cdots,n)$, that is, accept the best chromosome from another subpopulation and discard the worst chromosome of the subpopulation, and therefore yields the next population:

$$P_i(t+1) \leftarrow exchange \ [P(t), P_i'''(t)]$$

(7). Judge the ending conditions:

If the ending conditions are not satisfied, then $t \leftarrow t+1$, and transfer to execute step (4);

If the ending conditions are satisfied, then output the optimization solution and the execution of algorithm is over.

5. EXPERIMENT RESULT OF ALGORITHM PERFORMANCE

We have implemented the algorithm using the C language on the Linux platform to exam its performance. When the algorithm is being executed, the evolutionary process of each subpopulation corresponds to different child process. Two FIFOs in reverse directions, by which the information exchanges are realized, are established between each pair of child process, and consequently, the total number of the FIFOs in the system is $2C_n^2$. In our experiment, we input some task graphs, of which optimal scheduling is known, to test the efficiency of the PPGAs-based scheduling algorithm. These task graphs are varied in CCR. In the experiment, the initial values of p_c and p_m were set to 0.6 and 0.02, respectively. In addition, both k_c and k_m were also fixed at 0.6 and 0.02.

Table I shows outputs of the experiment and the percentage deviation from optimal Schedule Length for the selected task graphs, when k_p=2，3，5，10, k_g=20 and $n =$ 8. In the result table, the graph size of each line equals to the number of the nodes in the task graph. And simply put, the number of optimal scheduling and the average percentage deviation of each column represent the global and local search abilities of the genetic algorithm respectively. So they are two important indexes to evaluate the performance of the genetic scheduling algorithm.

Table II is the experiment result of PPGAs-Based scheduling algorithm, with other parameters not altered, by using swap operation as the mutation. In Table III, there is a group of data gained by SGA-Based scheduling algorithm with the same control parameters as those of the two former cases. In this testing experiment, the mutation operator is insertion operation.

Table I

k_p	2			3			5			10		
Graph	CCR											
Size	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	9.1	12.0	0.0	9.6	7.0	0.0	9.6	7.0	0.0	8.3	7.0
18	0.0	0.0	16.8	0.0	9.1	0.0	0.0	0.0	9.1	0.0	0.0	9.1
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	7.1	17.5	3.3	0.0	16.2	0.0	0.0	12.7	0.0	0.0	12.7
24	4.1	5.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	14.4	0.0	0.0	14.8	0.0	0.0	11.7	0.0	0.0	11.7
28	0.0	9.1	0.0	0.0	5.3	0.0	0.0	5.2	0.0	0.0	0.0	0.0
30	0.0	8.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	0.0
32	6.7	0.0	6.2	3.1	0.0	0.0	3.2	0.0	0.0	2.9	0.0	0.0
No.of Opt.	10	7	7	10	9	9	11	10	8	11	10	8
No.of Avg.dev.	5.4	7.7	13.4	3.2	8.0	12.7	3.2	7.4	10.1	2.9	6.7	10.1

Table II

k_p	2			3			5			10		
Graph	CCR											
Size	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	9.6	12.0	0.0	9.6	7.0	0.0	9.6	7.0	0.0	8.8	7.5
18	0.0	0.0	18.6	0.0	9.1	0.0	0.0	0.0	9.8	0.0	0.0	9.8
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	7.3	18.3	3.7	0.0	17.3	0.0	0.0	13.4	0.0	0.0	13.4
24	4.5	5,7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	15.2	0.0	0.0	15.2	0.0	0.0	12.5	0.0	0.0	12.5
28	0.0	9.1	0.0	0.0	5.3	0.0	0.0	5.7	0.0	0.0	0.0	0.0
30	0.0	8.7	0.0	0.0	5.0	0.0	0.0	0.0	0.0	0.0	5.5	0.0
32	7.2	0.0	6.8	3.5	0.0	0.0	3.5	0.0	0.0	3.3	0.0	0.0
No.of Opt.	10	7	7	10	9	9	11	10	8	11	10	8
Avg.dev.	5.9	8.1	14.2	3.6	7.2	13.2	3.5	7.9	10.7	3.3	7.2	10.8

Table III

k_p	2			3			5			10		
Graph	CCR											
Size	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0	0.1	1.0	10.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16	0.0	9.6	12.0	0.0	9.6	7.0	0.0	9.6	7.0	0.0	8.8	7.0
18	0.0	0.0	18.6	0.0	9.1	8.9	0.0	0.0	9.1	0.0	6.9	9.1
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
22	3.9	7.3	18.3	3.7	2.8	17.3	0.0	0.0	12.7	3.7	0.0	12.7
24	4.7	5.7	0.0	3.2	0.0	0.0	2.8	0.0	0.0	3.5	0.0	0.0
26	0.0	0.0	15.2	0.0	0.0	15.2	0.0	0.0	11.7	0.0	0.0	11.7
28	0.0	9.1	4.1	0.0	5.3	0.0	0.0	5.3	0.0	0.0	5.3	0.0
30	0.0	8.7	0.0	0.0	5.8	0.0	2.9	5.8	0.0	0.0	5.0	0.0
32	7.4	3.6	7.2	3.7	0.0	0.0	3.2	0.0	4.5	2.9	0.0	0.0
No.of Opt.	9	6	6	9	7	8	9	9	7	9	8	8
Avg.dev.	5.3	7.3	12.5	3.5	6.5	12.1	3.0	6.9	9.0	3.4	6.5	10.1

As for the task graphs in the previous experiment, we also tested the PPGAs scheduling algorithm with a larger number of

generations. The number of the subpopulations is also set to 8 and the other parameters remained the same. The results are shown in Fig.1.

Fig. 2 shows the results that we obtained from the testing experiment when the number of the subpopulation varies, k_g =40 and the other parameters remain the same as in the previous experiments.

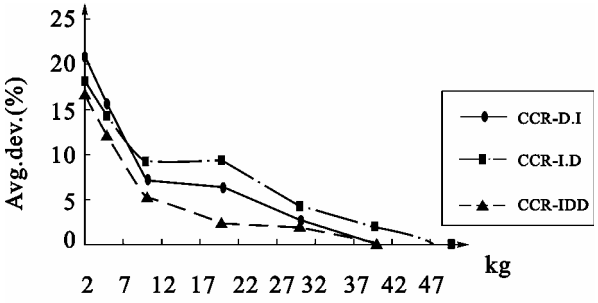


Fig.1

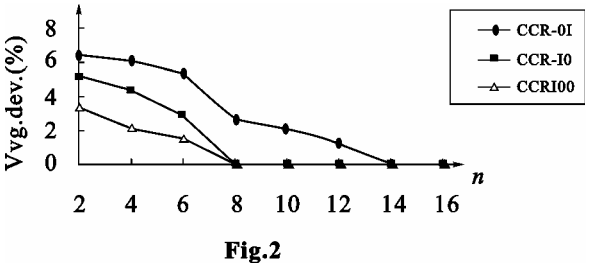


Fig.2

6. CONCLUSIONS

We have presented in the previous sections the pseudo-parallel genetic scheduling algorithm for distributed multiprocessor DAG scheduling. As can be see from the results of the performance-testing experiment, the insertion operation used as the mutation operator can actually enhance the local search ability of the genetic scheduling algorithm, while it has little influence on the global search ability of the algorithm. We also conclude that the PPGS algorithm has stronger global search ability than the SGA, because it can prevent premature.

Although the PPGAS has more efficient performance, further improvement are possible if we can determine an optimal set of control parameters, including crossover rate, mutation rate, population size, number of generations and number of the subpopulations. The integration of the genetic algorithm with other optimization search techniques, such as simulated annealing algorithms, is properly another road leading to the scheduling algorithms of higher efficiency.

7. REFERENCES

[1] Ishfaq Ahmad and Yu-Kwong Kwok, "On Parallelizing Multiprocessor Shceduling Problem," IEEE Transactions On Parallel and Distributed Systems, vol.10, no.4, pp. 414-431,April 1999.

[2] Ishfaq Ahmad and Yu-Kwong Kwok, "On Exploiting Task Duplication in Parallel Program," IEEE Transactions On Parallel and Distributed Systems, vol.9, no.9, 872-891, September 1998.

[3] Sekhar Darbha and Dharma P. Agrawal, "Optimal Scheduling Algorithm for Distributed-Memory Machines," IEEE Transactions on Parallel and Distributed Systems, vol.9, no.1, 87-94, January 1998.

[4] Yu-Kwong Kwok and Ishfaq Ahmad,"Efficient Scheduling of Arbitrary Task Graphs to Multiprocessors Using a Parallel Genetic Algorithm," Journal of Parallel and Distributed Computing 47, 58-77 (1997).

[5] S. Ali. S. M. Sait, and M. S. T. Benten, "GsA: Scheduling and Allocation Using Genetic Algorithm," Proceeding of EURO-DAC'94, pp.84-89.

[6] Sekhar Darba and Dharma P. Agrawal, "A Task Duplication Based Scalable Scheduling Algorithm for Distributed Memory Systems," Journal of Parallel and Distributed Computing 46, 15-27(1997).

[7] E. S. H. Hou, N. Ansari, and H. Ren, "A Genetic Algorithm for Multiprocessor Scheduling," IEEE Transactions on Parallel and Distributed Systems, vol.5, no.2, pp. 113-120, Feb. 1994.

[8] I. Ahmad and M. K. Dhodhi, "Multiprocessor Scheduling in a Genetic Paradigm," Parallel Computing, 22, 3 (Mar. 1996), 395-406.

[9] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, IEEE Transactions on Systems Man Cybernet. 24, 4(Apr. 1994), 656-667.

[10] A. Schoneveld, J. F. de Ronde, and P. M. A. Sloot, "Task Allocation by Parallel Evolutionary Computing," Journal of Parallel and Distributed Computing 47, 91-97 (1997).

[11] M. Srnivas and L.M. Patanik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms," IEEE Transactions on Systems Man Cybernet.24, 4 (Apr. 1994), 656-667mory Systems," Journal of Parallel and Distributed Computing 46, 15-27(1997).

Parallel Algorithm for ODE Modeling Problems

Lishan Kang, Yan Li , Yuping Chen, Hongqing Cao
Department of Computer Sciences, China University of Geosciences
State Key Laboratory of Software Engineering, Wuhan University
Wuhan 430072, P. R. China
E-mail: kang@whu.edu.cn

ABSTRACT

How to discover high-level knowledge such as laws of natural science in observed data automatically is a very important and difficult task in scientific research. In this paper, high-level knowledge modeled by systems of ordinary differential equations (ODEs) is discovered in observed dynamic data automatically by an asynchronous parallel evolutionary algorithm. A numerical example is used to demonstrate the potential of the parallel algorithm. The results show that the dynamic models discovered automatically in observed dynamic data by computer sometimes is comparable with the mode discovered by human.

Keywords: parallel algorithm, knowledge discovery, Evolutionary modeling

1. INTRODUCTION

A major impediment to scientific progress in many fields is the inability to make sense of the huge amounts of data that have been collected from a variety of sources. In the field of knowledge discovery in databases (KDD), there have been major efforts in developing automatic methods to find significant and interesting models (or patterns) in complex data and forecast the future based on those data. In general, however, the success of such efforts has been limited in the degree of automation during the process of KDD and in the level of the models discovered by data mining methods. Usually the goals of description and prediction are achieved by performing the following primary data mining tasks: summarization, classification, regression, clustering, dependency modeling, change and deviation detection [1]. Recently, Ngan, Wong, Leung and Cheng [2] used grammar based genetic programming for data mining of medical knowledge. Despite all those methods and models mentioned above, our research focuses on discovering high-level knowledge in complex data modeled by systems of ordinary differential equations (ODEs). We have ever proposed a two-level hybrid evolutionary modeling algorithm called HEMA [3, 4] to approach this task. In this paper, we parallelize it as an asynchronous parallel algorithm for suiting different computing systems, especially, the MIMD computers. Some numerical experiments were done to test its effectiveness. We use the parallel algorithm for modeling an example of the chemical reaction to demonstrate its potential in discovering the dynamic models in observed data automatically.

2. PARALLEL ALGORITHM FOR ODES MODELING PROBLEM

Suppose a dynamic system can be described by n interrelated functions $x_1(t), x_2(t), \cdots, x_n(t)$and a series of observed data collected at the time $t_i = t0 + i*\triangle t$ (i = 0,1,2, ,⋯,m –1)can be written as the following form

$$X = \begin{bmatrix} x_1(0), & x_2(0), & \cdots, & x_n(0) \\ x_1(t_1), & x_2(t_1), & \cdots, & x_n(t_1) \\ \vdots & \vdots & & \vdots \\ x_1(t_{m-1}), & x_2(t_{m-1}), & \cdots, & x_n(t_{m-1}) \end{bmatrix} \quad (1)$$

where t0 denotes the starting time (here $t_0 = 0$), △t denotes the interval between two observations, $x_j(t_i)(j = 0,1,2,\cdots,n)$ denotes the observed value of variable x_j at the time t_i.

If we denote $x(t)=[x_1(t), x_2(t), \cdots, x_n(t)]$, $f(t,x)=[f_1(t,x), f_2(t,x), \cdots, f_n(t,x)]$ where $f_j(t,x)= f_j(t, x_1(t), x_2(t), \cdots, x_n(t))(j =1,2, ,\cdots,n)$ is the composite function of elementary functions involving variables $x_i(i = 1,2,\cdots,n)$ and t , and the function space defined by those functions can be denoted as F, the modeling problem of system of ODEs is to find the model, a system of first-order differential equations having the general form of

$$dx^*/dt = f(t, x^*) \quad (2)$$

such that

$\min\{ \| X^* - X \| , \forall f \in F \}$

where X^* is matrix that is the values of $x^*(t)$ at the time points corresponding to matrix (1),

$$\| X^* - X \| = \sqrt{\sum_{i=0}^{m-1} \sum_{j=1}^{n} (x_j(t_i) - x^*_j(t_i))^2}$$

and the values of x_i(i = 0,1,2,⋯,n)at the next τ time steps

$$\begin{bmatrix} x_1(t_m), & x_2(t_m), & \cdots, & x_n(t_m) \\ x_1(t_{m+1}), & x_2(t_{m+1}), & \cdots, & x_n(t_{m+1}) \\ \vdots & \vdots & & \vdots \\ x_1(t_{m+\tau-1}), & x_2(t_{m+\tau-1}), & \cdots, & x_n(t_{m+\tau-1}) \end{bmatrix}$$

can be predicted based on the model(2).

To make sure of the validity of models, we assume that system of ODEs implied in the observed data satisfies some degree of stability with respect to the initial condition. Namely, the small change of the initial condition will not give rise to the great change of the solution of ODEs. The algorithm is the asynchronous parallel form of the HEMA (hybrid evolutionary modeling algorithm) we have proposed to approach the task of automatic modeling of ODEs for dynamic system. Its main idea is to embed a genetic algorithm (GA)[5] in genetic programming (GP)[6]-[8] where GP is employed to optimize the structure of a model, while a GA is employed to optimize the parameters of the model. It operates on two levels. One level is the evolutionary modeling process and the other one is the parameter optimization process.

Denote the population of ODEs by $P = \{p_1, p_2, \ldots, p_N\}$, where individual p_i is a system of ordinary differential equations represented by n parse trees. We assume that a population of N individuals is assigned to each of K processors. Each processor executes the same program PROCEDURE to steer the asynchronous parallel computation.

Procedure:
begin}
t := 0;
initialize the ODE model population P (t) ;

```
evaluate P ( t ) ;
while not (termination-criterion1) do
{ evolutionary modeling process begins}
 simplify P ( t );
 for k := 1 to MAX do}
{ MAX is the number of models chosen to optimize}
 choose p from P ( t )
 check out all the parameters contained in p;
 s := 0;
 initialize the parameter population P*(s );
evaluate P*(s );
while not (termination-criterion2) do}
      { parameter optimization process begins }
       s := s + 1;
    select P*(s ) from P*(s - 1 );
 recombine P*(s ) by using genetic operators;
 evaluate P*(s );
 endwhile { parameter optimization process ends }
 replace the parameters of p with the best individual in P*(s );
endfor
locate p_best and p_worst by sorting P ( t );
 if (t ≡ 0 (mod T )) then broadcast p_best to Q neighbors;
 while (any received message probed ) do
if ( recv— individual better than p_best ) then p_best:= recv-individual
 else p_worst := recv-individual;
 locate p_worst by sorting P ( t );
endwhile}
t: = t + 1;
select P ( t ) from P ( t—1);
recombine P ( t ) by using genetic operators;
handle the same class of models in P ( t ) by sharing techniques;
evaluate P ( t );
endwhile { evolutionary modeling process ends }
end
```

where $t \equiv 0 \pmod{T}$ denotes that t is congruent to zero with respect to modulus T.

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.

Once the best evolved model is obtained in one run, to check its effectiveness, we take the last line of observed data as the initial conditions, advance the solution of the ODEs model by numerical integration using some numerical methods, such as the modified Euler method, and get the predicted values for unknown data in the next time steps. As for the representation, fitness evaluation and genetic operators of these two processes, interested readers can refer to [4] to get more details.

3. NUMERICAL EXPERIMENT

The reaction between formaldehyde (X1) and carbamide in the aqueous solution gives methylolurea (X2) which continues to react with carbamide and form methylene urea (X3). The reaction equation [9] is

$$\begin{aligned} &HCHO + (NH_2)_2CO \xrightarrow{k_1} H_2N \cdot CO \cdot NH \cdot CH_2OH \\ &H_2N \cdot CO \cdot NH \cdot CH_2OH + (NH_2)_2CO \xrightarrow{k_2} \\ &(NH_2CONH)_2CH_2 + H_2O \end{aligned} \tag{3}$$

The reactions occur at 308.15K and under the excessive carbamide with the concentration ($c_{2(0)}$) of 2mol.dm^{-3}. The concentration of chlorhydric acid as catalyst is 0.0008mol.dm-3 and the initial concentration of formaldehyde ($X_{1(0)}$) is 0.1mol.dm^{-3}. As a kind of typical consecutive reaction, the concentrations of the three components in the system satisfy the following system of ODEs

$$\begin{cases} dX_1/dt = -k_1' X_1 \\ dX_2/dt = -k_1' X_1 - k_2' X_2 \\ dX_3/dt = k_2' X_2 \end{cases} \tag{4}$$

where $k_1' = k_1 c_{2(0)}, k_2' = k_2 c_{2(0)}$ with k_1=0.007dm^3.mol^{-1}.min^{-1}, k_2=0.021dm^3.mol^{-1}.min^{-1}. According to the exact solution of consecutive reactions

$$\begin{cases} X_1 = X_{1(0)} e^{-k_1' t} \\ X_2 = \dfrac{k_1' X_{1(0)}}{k_2' - k_1'}(e^{-k_1' t} - e^{-k_2' t}) \\ X_3 = X_{1(0)} - X_1 - X_2 \end{cases} \tag{5}$$

We calculate the concentrations of X_1, X_2, X_3 (every other minute) within 110 minutes since the reactions occur and take them as simulated data of our experiment (see [4]). Among them, the first 100 points are used as modeling samples and the next 10 points are used as test samples to evaluate the predicting results of the model.

The kinetic models discovered by computer are as follows:

(1)
$$\begin{cases} dX_1/dt = -[(11.44136\ 1t + 1/t)X_2 - X_1 + X_3]/2 \\ dX_2/dt = (\sin X_1/X_3 - 6.278076\ t)X_2/2 \\ dX_3/dt = (X_1 + 7.358359\ X_2 - 0.028639)/2 \end{cases}$$

(2)
$$\begin{cases} dX_1/dt = -1.4105651\ X_1 \\ dX_2/dt = (X_1 - X_2 - 0.036991)(1.92687 + X_2/X_3)/2 \\ dX_3/dt = (\ln \ln |X_1| /8.2341451)/2 \end{cases}$$

(3)
$$\begin{cases} dX_1/dt = -[((X_1^2+1)X_1 + X_2)/0.448129]/2 \\ dX_2/dt = (X_1 - (1+t)X_3)/2 \\ dX_3/dt = [(0.333855 - X_3)t + (1+t)X_1]/2 \end{cases}$$

(4)
$$\begin{cases} dX_1/dt = -1.3778625(1+X_3)X_1 \\ dX_2/dt = (2X_1 - 6.029049X_2)/2 \\ dX_3/dt = [X_1 + (0.709018 - t)X_2 + 0.308228t]/2 \end{cases}$$

(5)
$$\begin{cases} dX_1/dt = -1.3357545X_1 \\ dX_2/dt = [X_1 - X_3(1\text{-}X_3) + (t - 0.535889)(t - 0.159912)]/2 \\ dX_3/dt = [X_1 + (1.308533 + 15.167846t)X_2]/2 \end{cases}$$

These different forms of systems of ODEs are modeling the dynamics of the same chemical reaction equation (3) with almost the same approximate errors. They show that the computational intelligence can be competitive with the intelligence of mankind, even surpass it in some sense.

The numerical experiments are performed on a massively parallel computer system (an MIMD machine with 1024 nodes). Because the algorithm is performed in an asynchronous parallel

way, the processors need not wait for each other, and the linear speed-up is obtained.
The data in following table are the average values in ten runs:

node	CPU time	Speedup	Efficiency
1	119.4 s		
2	60.7 s	1.967	98%
4	33.9 s	3.5	88%
8	18.0 s	6.63	83%

4. CONCLUSIONS

In this paper, based on the hybrid evolutionary modeling algorithm HEMA [3,4] we propose an asynchronous parallel algorithm which can be run on the MIMD computers with PVM or MPI as the communication support and used to discover the kinetic models of chemical consecutive reactions. The experimental results show that by running the parallel algorithm, the computer can discover high-level knowledge modeled by a system of ordinary differential equations (ODEs) in observed dynamic data automatically. It can not only discover the dynamic model which can compare with the classical model or exact model, but also some suitable models whose structures are usually unimaginable for human. It is promising for the new algorithm to serve as a powerful parallel computing tool for the automatic discovery of the knowledge in dynamic data, especially for the discovery of scientific laws in observed dynamic data by parallel computing systems.

5. ACKNOWLEDGEMENT

This work was supported by National Science Foundation of China (No.60133010, No. 70071042 and No.60073043).

6. REFERENCES

[1] Fayyad, U.M, Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy,R. (eds.), Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, 1966.

[2] Ngan, P.S., Wong, M.L., Leung, K.S. , & Cheng, J.C.Y., Using grammar based genetic programming for data mining of medical knowledge. In Genetic Programming 1998: Proceedings of the Third Annual Conference, David E., Iba, Hitoshi, and Riolo, Rick L. (editors). pp.254-259. University of Wisconsin, Madison, Wisconsin. San Francisco, CA, July 22-25, 1998: Morgan Kaufmann.

[3] Cao,H.Q., Kang,L.S., Michalewicz, Z., &Chen, Y. P., A Hybrid evolutionary modeling algorithm for stem of ordinary differential equations. Neural , Parallel & Scientific Computations. Vol.6, No.2, pp. 171-188, June, 1998, Dynamic Publishers, Atlanta.

[4] Cao,H,Q., Kang,L.S., Chen,Y.P.,& Yu,Z.X.,. Evolutionary Modeling of Systems of Ordinary Differential Equations with Genetic Programming. Genetic Programming and Evolvable Machines. Vol.1, No.4, 2000, 309-337.

[5] Mitchell, M., An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press, 1996.

[6] Koza, J. R., Genetic Programming: on the Programming of Computers by Means of Natural Selection. Cambridge, MA:MIT Press, 1992.

[7] Koza, J. R., Genetic Programming II: Automatic Discovery of Reusable Programs. Cambridge, MA:MIT Press, 1994.

[8] Koza, J.R., Bennett, F.H, Ⅲ; Andre,D., Keane, M. A., Genetic Programming Ⅲ: Darwinian Invention and Problem Solving, San Francisco, Morgan Kaufmann, 1999.

[9] Moore, J. W. & Pearson, R. J., Kinetic and Mechanism: A Study of Homogeneous Chemical Reactions (Third Edition). John Wiley, 1981.

ACER：Alternating Cyclic Elimination and Reduction Algorithm For the Solution of Tri-diagonal Systems

Hai Xiang Lin, Jarno Verkaik
Department of Applied Mathematics
Faculty of Information Technology and Systems
Delft University of Technology
2628 CD, Delft, The Netherlands
E-mail: h.x.lin@its.tudelft.nl

ABSTRACT

In this paper we use the unifying graph model to design a class of new parallel algorithms for the solution of tri-diagonal matrix systems. The new algorithms, called ACER (Alternating Cyclic Elimination and Reduction),combine the advantages of the well known cyclic elimination algorithm (which is fast) and the cyclic reduction algorithms (which requires fewer operations).

Keywords: parallel matrix algorithm, graph transformation, unifying graph model, ACER algorithms.

1. INTRODUCTION

The solution of tri-diagonal systems occurs in many engineering and scientific computer applications, for instance, it is often a part of the solution process in numerical simulation of PDEs using finite difference or finite element discretization. On the other hand, when standard Gaussian elimination method is applied to a (pre-ordered) tri-diagonal matrix, the computation is inherently sequential. That is why it has inspired many researchers to study parallel algorithms or solving this problem (in the past three decades more than 200 journal papers have been published on this problem , e.g., see an online list of literatures[8]).

Because of the serial data dependence in the standard Gaussian elimination/factorization process prohibits any parallelization, so different algorithms are designed which trade doing extra arithmetic operations for a higher degree of parallelism. Such trade-of is typical in designing parallel algorithms for problems whose efficient parallelization are not straightforward. The solution of tri-diagonal system is such a problem that ideally shows the trade-off considerations in designing algorithms for not so straightforward or hard to parallelize problems.

As we have mentioned that during the last four decades a large variety of parallel algorithms for the solution of tri-diagonal systems have been proposed, some well known examples are: the cyclic reduction algorithm [3,5], the cyclic elimination, the recursive doubling algorithm [13], the block partitioned elimination algorithms [4,11,14], and the block portioned (Cholesky) factorization algorithms [7]. These and other algorithms are designed in an ingenious way and by different researchers in parallel computing through the years. The different approaches are often presented in a different way like partition of rows or columns of the matrix, index permutation and elimination tree, etc. recently, a unifying graph framework has been presented [9] which not only can describe many of these algorithms in a unified and comprehensive way but also can be used to design new algorithms with certain desired property.

In this paper we present a class of new algorithms, called ACER (Alternating Cyclic Elimination and Reduction algorithm), which combine the advantages of the well known cyclic elimination and the cyclic reduction algorithm. In the following, first the unifying graph theoretic framework is introduced in Section 2, then in Section 3 we present a new class ACER algorithms designed using the graph framework. Analysis show that the ACER algorithms are faster than the cyclic reduction algorithm, and nearly as fast as the cyclic elimination algorithm, but they require approximately the same number of operations as the cyclic reduction algorithm which is asymptotically a factor $\log_2(n)$ fewer than the cyclic Elimination algorithm. In Section 4 concludes the paper with some remarks and open questions.

2. THE GRAPH THEORETIC FRAMEWORK

In this section we introduce the graph theoretic model for the parallel elimination of a general matrix. First the relationship between graph transformation and Gaussian elimination for a general matrix is reviewed. We then proceed the discuss the basic relations between parallel elimination of edges and fill-ins[1]. A number of properties of graph transformation and parallel elimination will be investigation, elimination and update operations which are used to transform an initial graph into a graph with the required property.

The undirected elimination graph has been successfully used for minimizing fill-ins in sequential solution (e.g.[1,2]) and parallel factorization of sparse symmetric matrices. However, it cannot describe many operations in a parallel matrix algorithm. For instance the recursive doubling algorithm [13] and the partition method [14] cannot be described in terms of elimination each column or row exactly once during the elimination process. In the partition method some row or columns are modified several times. Unlike in Gaussian elimination of LU-factorization the final form here is not an upper triangular matrix. Therefore, we use directed graphs in our framework, In order to obtain the required high flexibility we study the parallelism in elimination an arc (j, i) associated with a (j, i) using a (i, i) instead of elimination an entire column i. Note that we don't assume $i > j$ or $j > i$ here, i.e., we don't assume any pre-determined elimination ordering. We consider the elimination of an arc as an operation which can be performed in any order or even repeatedly. Throughout the paper, we ignore numerical cancellations when considering fill-ins and fill-arcs during an elimination process. A non-zero is thus a logical non-zero and an accidental numerical cancellation is not considered as a zero coefficient.

Given an $n \times n$ matrix A, a corresponding directed graph G (V, E) is defined as a graph with the set of nodes $V=\{1,2,\cdots,n\}$ and the set of arcs $E=\{(i,j)\} \mid a(i,j) \neq 0, i \in V \wedge j \in V\}$. Arc (i, j) is said to be have a begin node i and end (or terminal)

[1] A fill-in is a coefficient which is zero in the original matrix A,, but becomes non-zero during the elimination/factorization.

node *j*. (*i, j*) is an outgoing arc from *i*, and an incoming arc to *j*. The set of predecessors and successors of node *i* are denoted by PRED(i) and SUCC(*i*). As a convention, in this paper we define that the node *i* is never a predecessor of successor of itself.

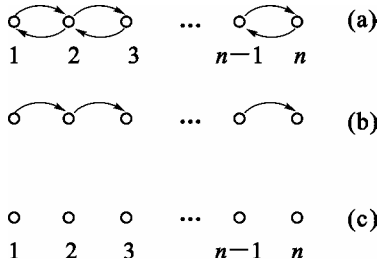


Figure 1 Graph transformations corresponding to the Gaussian elimination

Theorem 1 [9] Consider the elimination of arc (*i,j*) using node *j*, arc (*i,k*) exists if and only if (*i,k*) or (*j,k*) exists before the elimination.

The graph corresponding to a tridiagonal matrix is a bidirectional linear chain (Fig. 1a). The standard Gaussian elimination algorithm eliminates edge (*k+1, k*), *k*=1,2,⋯*n*-1, at step *k* during the (forwards) elimination process. After the (forwards) elimination, a graph with a path staring from node *n*, through *n-1*, ⋯, to node 1 results(Fig. 1b). in this graph there are only edges (*i, j*) with *i=h* or *j =i*+1, this corresponds to an upper triangular bidiagonal matrix. The unknown can now thus be computed thorough back-substitution, starting from *n*, through *n*-1,⋯, to 1. This result in a graph of *n* disconnected nodes (which correspond to a diagonal matrix). Notice that the standard Gaussian elimination process is inherently sequential, only one arc at a time is eliminated.

The corresponding numerical operations on the matrix can be defined on this graphical representation. We denote the coefficient a (*i, j*) as the weight of arc (*i, j*), a non-existing arc is equivalent to an arc having zero weight. The elimination of (arc (*i,j*) will correspond to the following operations in the graph:

- for each $k \in SUCC(j)$: $a(i,k) = a(i,k) - a(i,j) * a(j,k)/a(j,j)$. (Note that if $a(i,k)$ was 0, i.e., arc (i,k) does not exist, then (i,k) is a fill-arc).
- remove arc (i,j);

Since we are only interested in the parallelism and the structure of the parallel operations in this paper, we will omit the discussion of the numerical operation of the coefficients. The weight of the arcs is also omitted except htat we define there is an arc when a (i,j) is logically non-zero, and (*i,j*) does not exit when a (*i,j*)=0.

Lemma 1.1 *The elimination of arc (i,j) results in arcs (i,k) for all successors of j, i.e., after the elimination we have (i,j) for each k∈SUCC(j). These are the only modifications in the edge-connectivity as a result of the elimination of (i,j).*

Lemma 1.1 follows straightforward from Theorem 1. It states that a fill-arc *(i,k*) occurs when eliminating arc (*i,j*) if and only if there is a path i → j → k, and *(i,k*) does not exist before the elimination of (*i,j*). Consequently the elimination of all incoming arc to node *i* will result in all predecessors of *i* connected to all successors of *i*. The elimination of all outgoing arcs (*i,j*) from node *i*, result in node *i* become connected to all nodes in SUCC(*j*) for reach *j*∈SUCC(*i*). In matrix terminology, the elimination of all incoming g and outgoing arcs of node *i* corresponds to the elimination of all off-diagonal non-zeros of column in and row i respectively. Lemma 1.1 also tell sus the so-called 'path-conservation' principle of the transformation: of k_1 and k_2 are connected then k_1 and k_2 remains connected when an arbitrary arc (*i,j*) in the graph is eliminated provided that $(k_1,k_2) \neq (i,j)$.

Lemma 1.2 *Parallel elimination of (i_1,j_1) and (i_2,j_2) is successful if and only if*

1. $i_1 \neq i_2$; *or*
2. $i_1 = i_2$ *and* j_1 *is not in* $SUCC(j_2)$ *and* j_2 *not in* $SUCC(j_1)$.

Lemma 1.2 can be proved directly using Theorem 1. We define a parallel elimination step as the elimination step as the elimination of a set of arcs which can be eliminated independently. A parallel elimination of two acres is said to be successful if they are both eliminated after the parallel elimination step. This is of course not always possible because the elimination of arc (i_2,j_2) may cause the return of (i_1,j_1) (although now with another value of $a(i_1,j_1)$), and vice versa. Lemma 1.2 tells us that parallel eliminations of two arcs initialing from the same begin node can only be successful if there is no arc between the terminal nodes of these two arcs. Take for example Fig.2, the set of arcs {(1,2), (2,1), (2,3), (3,2, (4,3)) can be successfully eliminated in parallel, but the pair (1,2) and (1,4) cannot be eliminated in parallel. [2]This conclusion can easily be extended to a set of arcs.

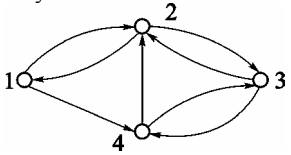


Figure 2 An example graph

Consider a directed graph G(V,E) associated with an $n \times n$ matrix. An algorithm, which determines the parallel elimination of arcs in G until all nodes become isolated is given in Algorithm 1. The elimination of the arcs in each set S (*k*) comprises one elimination step. Note that other possible end conditions are: each node in the graph has only either outgoing or incoming arcs, the remaining graph forms a single or multiple spanning tree(s). This is a fundamental difference form the case of undirected graphs where the end configuration is always a triangular matrix (corresponding to a nodes are eliminated from the elimination graph). Another fundamental difference between the presented directed graph model and the conventional undirected graph is that in case of a directed graph mode la parallel elimination step is generally not divisible into two of more equivalent substeps (as will be discussed later in the paragraph about update conflicts).

```
Initialize G(V,E)with V={1,2,…,n} and
          E={{I,I+1}\i=1,…,n-1}∪
            {(I,I+1)\i=2,…,n};
k=1;
WHILE E≠0 DO
    SELECT all possible arcs S(k) for parallel
          Elimination (Lemma 1.2);
    ELIMINATE s(k) from E;
    UPDATE E according to Lemma 1.1;
    K=k+1;
    END
```

Figure 3 Agorithm 1: A greedy algorithm for parallel

[2] they can be eliminated in the order of first (1,4) and then (1,2), but not first (1,2) followed by (1,4), in the latter case the arc (1,2) will return.

elimination

The key is the Selection operation. Algorithm 1 is a greedy algorithm in which at each parallel elimination step as many as possible arcs are selected for parallel elimination. Fig.4 shows that the application of the greedy algorithm to a bi-diagonal system os equations (linear recurrence) resulting in a parallel algorithm which is the known recursive doubling algorithm [13] for first order linear recurrence. At the first elimination step, all arcs (i+1, i) for i=1,2,⋯,(n-1) can be eliminated in parallel. According to Lemma 1.1 the elimination of (i+1,i) causes a fill-arc (i+1,i-1). In the second elimination step, all arcs (i+1,i-1) are eliminated in parallel and resulting in fill-arcs (i+1,i-3), etc. In general, at elimination step k, all arcs (i+1,i-2^k+1) are eliminated in parallel and fill-arcs (i+1,i-2^{k+1}+1) are added. This is exactly what the recursive doubling algorithm does for a bidiagonal system.

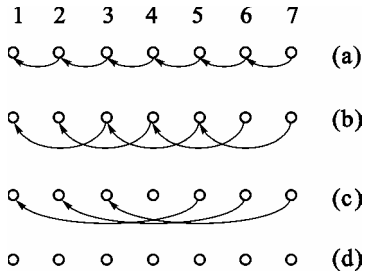


Figure 4 Application of Algorithm 1 to a bi-diagonal system leads to the recursive doubling scheme

Fig.5 shows the results of applying the algorithm to an example. Compared to the cyclic reduction algorithm, it can be observed that the number of parallel elimination steps achieved with the greedy algorithm is smaller. The time complexity of the greedy algorithm is $4^{\log_2}(n)+1$ versus $8^{\log_2}(n+1)-8$ for the cyclic reduction algorithm. This time complexity is obtained if maximum parallelism in execution is exploited, i.e., each update of the modified coefficient will be executed in parallel (whenever data dependence allows). If the update of each coefficient is done by 1 processor only, then the time complexity of the greedy and the cyclic reduction algorithm is $6^{\log_2}(n)+1$ and $11^{\log_2}(n+1)-12$ respectively. In the following we assume maximum parallelism in execution for time complexity analysis unless specified otherwise. We notice that in some of the steps of the greedy algorithm the elimination of several arcs initiating from the same node causes update conflict. A parallel update conflict occurs when the same coefficient is modified/updated more than once in a parallel elimination step. For example, the elimination of the pair of arcs (2,1) and (2,3) in Fig.2 has conflict in the update of the coefficient a (2,2).

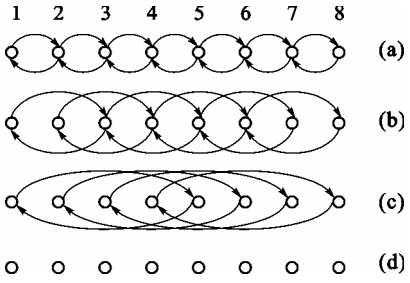


Figure 5 Illustration of applying Algorithm 1 to a tridiagonal sysyem

The next theorem states the conditions when the parallel elimination of arcs (i_1,j_1) and (i_2,j_2) is free of update conflict.

Theorem 2 [9] The elimination of arcs (i_1,j_1) and (i_2,j_2) are free of update conflict if

1. $i1 \neq i2$;or
2. if i1=i2 and $SUCC(j1) \cap SUCC(j2)=0$

If we require each parallel elimination step the elimination of arcs to be free of update conflict, we can add the above update conflict test in the selection operation of Algorithm 1. That is *"SELECT a set of arcs S(k) where for each pair (i_1,j_1) and (i_2,j_2) in the set conditions of Lemma 1.2 and Theorem 2 are satisfied."* Applied with this modification to the example in Fig.5, the result of parallel elimination steps is increased to 5 in Fig.6 compared to 3 in Fig.5. However, there are update conflicts in the first two steps in Fig.5 which implies a larger parallel execution time in these two steps.

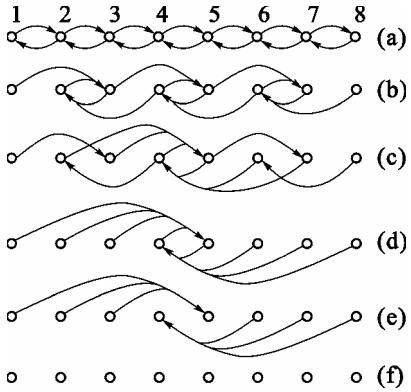


Figure 6 Illustration of a parallel elimination scheme free of update conflict

Furthermore, we observe that by simply the first step of the algorithm in Fig.5 into two sub-steps without update conflict, e.g., first eliminate all arcs (i,i+1) and then all arcs (i,i-1), will result into a totally different (in this case unfavorable) algorithm. In general, a splitting of a parallel elimination step into more sub-steps will result in a different parallel algorithm. This is essentially different from the case of parallel factorization of a symmetric matrix modeled using undirected graphs(e.g.[6]), where the split of a parallel factorization step into several (sequential) sub-steps results in the same algorithm (i.e. ,fill-ins and update operations are unchanged).

3 THE ACER ALGORITHMS

The tests of parallel elimination (Lemma 1.2J) and the test of update conflict (Theorem 2) can be simplified by allowing each node being an initiating node at most once in a single parallel elimination step. Furthermore, additional conditions can be limiting the number of fill-arcs, or requiring the total number of fill-arcs must be smaller than the number of arcs being eliminated in each step (the latter is to ensure the finiteness of the elimination process). Additional conditions or heuristics in the selection can also be used for more regularity and control on the parallelism. In fact many of the known parallel algorithms in literature corresponds to applying a certain heuristics or imposing some structure in the elimination process. The general algorithm can be described as consisting of three basic type of operations: 1. partition; 2. selection; and 3. elimination and update.

In the following we use the framework to design a class of new algorithms, the ACER (Alternating Cyclic Elimination and Reduction) algorithms. The ACER algorithms combine the advantages of the cyclic elimination and the cyclic reduction algorithms. First we describe the cyclic reduction and cyclic elimination algorithms using the unifying graph

model.

3.1 The cyclic reduction algorithm

Let the dimension of the matrix $n=2^k-1$.Consider a set of three consecutive equations, centered around i=2,4,6,8,…,n-2. The basic idea of this type of algorithm [3] is to use the odd numbered equations, i.e., equations (i-1) and (i+1) to cancel the variable $x_{i-1}$and x_{i+1} in the i-th equations, resulting in $(\frac{n}{2}-1)$ equations with only even numbered variables. This is described in the following.

$$a^{(0)}_{i-1,i-2}x_{i-2}+a^{(0)}_{i-1,i-1}x_{i-1}+a^{(0)}_{i-1,i}x_i = b^{(0)}_{i-1}(1)$$

$$a^{(0)}_{i,i-1}x_{i-1}+a^{(0)}_{i,i}x_i+a^{(0)}_{i,i+1}x_{i+1} = b^{(0)}_{i}(2)$$

$$a^{(0)}_{i+1,i}x_i+a^{(0)}_{i+1,i+1}x_{i+1}+a^{(0)}_{i+1,i+2}x_{i+2} = b^{(0)}_{i+1}(3)$$

Multiplying Eq. (1) and Eq . (3) with $-a_{i,i-1}/a_{i-1,i-1}$and $-a_{i,I+1}/a_{i+1,I+1}$ respectively ,and adding them to Eq. (2), we eliminate the variables x_{i-1} and x_{i+1} from Eq. (2), resulting in

$$a^{(1)}_{i,i-2}x_{i-2}+a^{(1)}_{i,i}x_i+a^{(1)}_{i,i+2}x_{i+2}=b^{(1)}_i \qquad (4)$$

The cyclic reduction algorithm consists of two phases (Ⅰ) the reduction phase, during which selective elimination of is done; and (Ⅱ) the back-substitution phase, during which the values of the eliminated x_i's are recovered.

Using the framework, we can now generate the cyclic reduction algorithm as shown in Fig. 7. The parallel elimination proceeds by starting with the elimination of all arcs initiated from even numbered nodes in the first step, followed by repeatedly eliminates the set of arcs initiated at even numbered nodes with a distance of 2^{k-1} to the end node at step k (they are independent from each other). Fig. 8 illustrates the different elimination steps in the reduction and back substitution phase of the reduction algorithm.

```
/*reduction phase*/
FOR m=1 TO log2(n+1)−1 DO
  h=2^(m-1)
  SELECT S(m)={(2ih,2ih-h),(2ih,2ih+h)|
        i=1,…,(n+1)/2n−1}
  for parallel elimination;
  ELIMINATE the arcs in S(m) form the graph;
  UPDATE;add fill-arcs {(2ih,2ih−2h),
      (2ih,2ih+2h)|i=1,…,(n+1)/2n−1};
END
/*back substitution phase*/
FOR m=log2(n)−1 TO 1 DO
  h=2^(m-1);
SELECT S(m+log2(n+1)−1)={(2ih−h,
   2ih−2h),(2ih−h,2ih)|i=1,…,(n+1)/2n−1};
ELIMINATE arcs in S(m+log2(n+1)−1)
      form the graph;
UPDATE:there are no fill-arcs;
END
```

Figure 7 Algorithm 2: Te cyclic reduction algorithm in terms of the framework. $n=2^k-1$

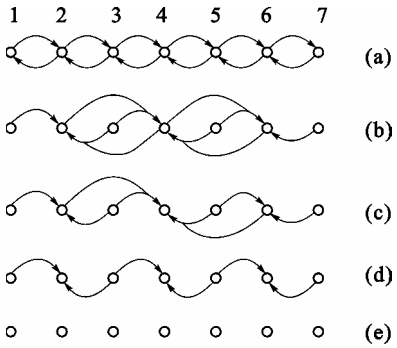


Figure 8 Illustration of the graph transformation process corresponding to the cyclic reduction algorithm.(b)-(c):the reduction phase;(d)-(e):the back-substitution

3.2 The cyclic elimination algorithm

The cyclic elimination algorithm repeatedly eliminates the arcs ($i,i+h$) and ($i,i-h$) with $h=2^{m-1}$ at step m. This algorithm can be directly generated by the greedy algorithm (Fig. 3), the graphical illustration of the cyclic elimination algorithm is shown in Fig. 5.The cyclic elimination algorithm reduces the tridiagonal system to diagonal system in $log_2(n)$ elimination steps each consisting of 4 floating-point executions(assume that equations(5) ,(6),(7) and (8) are executed in parallel). So the parallel time complexity of the cyclic eliminations is $4log_2(n)+1$. The time complexity is smaller than the cyclic reduction algorithm which has a time complexity of $8log_2(n+1)-8$.The price to be paid is the additional fill-arcs and thus a larger total number of floating point operations(when counted sequentially).

3.3 The ACER algorithm

As we have observed in the previous sections, the cyclic elimination algorithm uses a greedy approach in elimination the arcs in parallel which has a smaller time complexity of $4log_2(n)+1$ compared to $8\log_2(n+1)-8$ for the cyclic reduction algorithm. When we look at the total number of floating point operations or the number of fill-arcs, we see that the number of fill-arcs for the cyclic reduction algorithm is $2(n-1)-4\log_2(\frac{n+1}{2})$ which is much smaller than $2n\cdot\log_2(n)-4n+4$ for the cyclic elimination algorithm. This brings us to the combining the advantages of these two algorithms. Using the graph framework we can easily visualize and study the different variants of parallel elimination algorithms, and that results in an ACER algorithm which performs alternating cyclic reduction and cyclic elimination until the entire matrix system is solved. Fig. 9 describes the ACER algorithm in terms of graph transformation using the framework.

The ACER algorithm in Fig.9 contains $\log_l(n+1)$ reduction phases, l is the basic partition size of the matrix where the cyclic elimination process is applied. Within each reduction phase t, cyclic eliminations are performed for the partitions (submatrices) of $h_1=l^{t-1}$ rows, followed by a single reduction step at the end of each phase. Notice that the ACER algorithm is flexible in taking the values of the matrix dimension n, which can be $n=2l^k-1$ with l=2, 3,… And k=1,2,3… This in contrast to the limited values for both the cyclic elimination ($n=2^k-1$) algorithms. By choosing different base l, we can obtain a variety of variants of ACER algorithms. Notice that when l=2, this special case of the ACER algorithms is identical as the cyclic elimination algorithm.

The time complexity of the ACER algorithms[3] is $4m_1+8-1_{[m_1>0]})\log_l(\frac{n+1}{2})$ and the number of fill-ins is $\{2m_1-\frac{2^{m_1+1}-4}{l-1}\}(n-1)-2l\cdot\log_l(\frac{n+1}{2})$, where $m_1=\log_2(l-1)$ and $1_{[m_1>0]}$ is an indicator function, $1_{[m_1>0]}=1$ if $m_1>0$; =0 otherwise. Take for example l=3, then

[3] If the update of each coefficient is done by 1 possessor only (instead of maximum parallelism) then the time complexity of the ACER algorithm is $6m_1+11-3\cdot1_{[m_1>0]})\log_l(\frac{n+1}{2})\cdot$

the time complexity of the ACER algorithm is approximately $6.9\log_2(n+1)$ which is larger than $4\log_2(n)+1$ for the cyclic elimination algorithm but smaller than the time of $8\log_2(\frac{n+1}{2})$ for the cyclic reduction algorithm.Compaing the number of fill-ins, the ACER algorithm has approximately ***fill-ins which is a factor of $\log_2(n)$ smaller than the cyclic elimination algorithm and about the same as the cyclic reduction algorithm. Thus the ACER algorithms combine the advantages of the cyclic elimination and cyclic reduction algorithms.

4 CONCLUDING REMARKS

We have described a general graph theoretic framework which unifies many different parallel algorithms for sparse matrix computations. Using this framework we have designed a class of new parallel algorithms, the ACER algorithms , for the solution of tri-diagonal matrix systems. It has been shown that the ACER algorithms combine the advantages of both the cyclic elimination and cyclic reduction algorithms. The power of the graph framework lies in that we can design parallel algorithms beyond the ability of detecting parallelism in an existing algorithm [10]. It remains an interesting and challenging problem to define similar frameworks for sparse matrix computations other than the Gaussian-elimination and factorization.

We believe many other parallel algorithms can be designed and analyzed using the graph framework. An interesting open question is : what is the minimum parallel time complexity to solve a tri-diagonal system? For the tri-diagonal matrix systems, so far the fastest algorithm has a parallel time complexity of $O(\log_2(n))$ (actually $4\log_2(n)$). Can we use the graph model to prove the lower-bound of parallel time complexity for a tri-diagonal system? (under the given context of the set transformation rules).

5 REFERENCES

[1] I.S.Duff,A.M. Erisman and J.K. Reid, Direct Methods for Sparse Matrixs (Clarendon Press, Oxford ,1986).

[2] A. George and J.W.H. Liu, Computer solution of large sparse positive definite systems (Prentice-Hall,1981).

[3] R.W.Hockney, A fast direct solution of Poisson's equation using Fourier analysis, Jouranal of ACM 12(1965) 95-113.

[4] S.L.Johnsson, Solving Tridiagonal Systems on Ensemble Architectures, SIAM J. Sci. Stat. Comput. 8(1987)354-392.

[5] J.J.Lambiotte and R.G.Voigt, The solution of tridiagonal linear systems on the CDC STAR-100 computer, ACM TOMS 1 (1975)308-329.

[6] H.X.Lin, A methodology for the parallel direct solution of finite elements , Ph.D. thesis, (Delft University of Technology 1993).

[7] H.X.Lin and M.R.T.Roest , Parallel solution of symmetric banded systems, in: Parallel Computing: Trends and Applications, G.R.Joubert, D.Trystram, F.J.Peters and D.J. Evans(eds.), (Elsevier Science.1994(537-540.

[8] A bibliography on Parallel Solution of Tri-diagonal Systems of Equations. //ta.twi.tudelft.nl/wagm/users/lin/tri_sol.html

[9] H.X.Lin A Unifying Graph Model for Designing Parallel Algorithms For Tridiagonal Systems, Parallel Computing, Vol.27, 2001, pp 925-939.

[10] H.X.Lin Design Parallel Sparse Matrix Algorithms Beyond Data Dependence Analysis, Proc.ICCP 2001,Workshop High Performance Scientific Engineering Computing with Applications (Keynote), IEEE Computer Society Press, Valencia, Sept 3-7,2001.pp 7-13.

[11] U. .Meijer, A parallel partition method for solving banded systems of linear equations, Parallel Computing 2(1985) 33-43.

[12] S. Parter, The use of linear graphs in Gaussian elimination, SIAM Review 3(1961) 119-130.

[13] H. S. Stone, An efficient parallel algorithm for the solution of a tridiagonal linear system of equations, Journal of ACM 20(1973)27-38.

[14] H.H. Wang, A parallel method for tridiagonal equations, ACM TOMS 7(1981) 167-183.


```
/*reduction phase*/
FOR r=1 TO log_f((n+1)/2) DO
  h_1=2^(r-1)
  /*conform cyclic elimination*/
  FOR q=1 TO m_1 DO
    h_2=2^(q-1);
  SELECT S(q+(m_1+1)(r-1))=
       {((i-1)lh_1+jh_1,(i-1)lh_1+jh_1+h_1h_2),
       ((i-1)lh_2+jh_1+h_1h_2,(i-1)lh_1+jh_1)
       |i=1,…,(n+1)/(lh_1),j=1,…,l-1-h_2}
       for parallel elimination;
  ELIMINATE arcs in S(q+(m_1+1)(r-1)) form graph;
  UPDATE: add fill-arcs
       { ((i-1)lh_1+jh_1,(i-1)lh_2+jh_2+2h_1h_2),
       ((i-1)lh_1+jh_1+h_1h_2,(i-1)lh_1+jh_1-h_1h_2)
       |i=1,…,(n+1)/(lh_1),j=1,…,l-1-h_2};
  END
  /*conform cyclic reduction*/
  SELECT S(m_1+1)r)={(ilh_1,ilh_1-h_1),
       (ilh_1,ilh_1+h_1)|i=1,…,(n+1)/(lh_1)-1}
       for parallel elimination;
  ELIMINATE arcs in S(m_1+1)r) from the graph;
  UPDATE: add fill-arcs{(ilh_1,ilh_1-lh_1),
       (ilh_1,ilh_1+lh_1),|i=1,…,(n+1)/(lh_1)-1};
  END
  /*back substitution phase*/
  FOR r=log_f((n+1)/2 TO 1 DO
    v=2^(r-1);
    SELECT S((m_1+2)log_f((n+1)/2)-r+1)=
         {ilv+jv,ilv),(ilv-jv,ilv),|i=1,…,
         (n+1)/(lv)-1,j=1,…,l-1}
         for parallel elimination;
    ELIMINATE arcs S((m_1+2log_f((n+1)/2-r+1)
         from the graph;
    UPDATE: there are no fill-arcs;
  DND
```

Figure 9 Algorithm 3: The ACER algorithm $n=2l^k-1$,$l=2,3,\ldots$,$k=1,2,\ldots$,and $m_1=\log_2(l-1)$

Deconvolution Algorithms For Coincidence Doppler Broadening Spectra On PC Cluster *

Michael Ng
Department of Mathematics, The University of Hong Kong
Pokfulam Road, Hong Kong
E-mail: mng@maths.hku.hk

And

King Fung Ho, Vincent Cheng and Chris Beling
Department of Physics, The University of Hong Kong
Pokfulam Road, Hong Kong

And

Chat Ming Woo
Computer Center, The University of Hong Kong
Pokfulam Road, Hong Kong

ABSTRACT

Recent years have seen a renewed interest in the technique of Coincidence Doppler Broadening Spectroscopy (CDBS) in which one-dimensional electronic momenta in materials are studied by means of the energies of the two gamma-rays emitted in the process of positron annihilation. Advantages of CDBS over conventional positron Doppler spectroscopy are its 40% improved resolution and its much reduced background noise at high momenta. The present work capitalizes on the fact that CDBS raw data is in the form of a very large two-dimensional image, with excellent prospects for designing parallel deconvolution algorithms for the removal of the instrumental error of measurement that arises from the availability of an accurate point spread function in the reference gamma-ray line of Sr at 514keV. The generalized least-square method with Tikhonov-Miller regularization is developed by incorporating a priori information of non-negativity into the mathematical regularization technique for the solution of blurring matrix equations. This paper reports the performance of the parallel image deconvolution algorithm on the PC Cluster.

1. INTRODUCTION

Many studies of positron interactions with electrons in condensed matter have been developed into useful analytical techniques in recent decades. The positron, which is generally described as a non-destructive probe of electronic structure, annihilates with an electron in the condensed matter giving information on the momentum of the electron through the annihilation quanta released in the process. A consequence if the annihilation of positrons from either the delocalized, trapped or positronium states gives rise to observable that are now generally classified under the general heading of Positron Annihilation Spectroscopy: Positron Lifetime Spectroscopy (PLTS), Doppler Broadening Spectroscopy (DBS) and Angular Correlation of Annihilation Radiation (ACAR). In PTLS, information comes from the longer defect related lifetime component in the spectra. In DBS and ACAR, it is electron momentum distribution that is investigated. However, DBS technique suffers from a poorer momentum resolution than its 1D ACAR counterpart. In order to deal with this demerit, a pair of coincidence detectors has in recent years been re-introducing to reduce background and to sharpen the instrumental response function. This technique, named as Coincidence Doppler Broadening Spectroscopy, allows one to study sensitive core annihilation and thus the chemical environment of the positron within the defect as a result of extremely low random background at high momentum [5].

In regard to the poorer resolution of DBS, CDBS has the intrinsic property of improving the instrumental resolution by a factor of $\sqrt{2}$ in Doppler energy spectroscopy of the 511keV line [6]. Britton [3] demonstrated this by building a system, which after deconvolution had a 386 eV effective resolution is about 1.5 mrad (ACAR equivalent) and pinpointed that further improvements by deconvolution could make the resolving power of CDBS competitive to ACAR spectrometers. The objective of this paper is to explore this feasibility by deconvoluting the spectra of CDBS to a level that the resolution is competitive with ACAR.

A major factor in the success of any deconvolution venture is the quality of the input spectrum itself. Assuming the spectrum has been perfectly stabilized against electronic drift effects, there are still the uncertainties due to noise arising from the stochastic nature of the counting process. Thus the more counts in the spectrum, the more true to the convoluted functional shape it becomes. A total 511 keV peak energy and 10^9 events can be recorded by introducing at a rate of 10^4 throughout a day [2]. With a modern nuclear analogue to digital converter digitizing into a large number (e.g., 16000) of channels, a satisfactorily high signal-to-noise ratio spectra can thus be achieved. But this is problematic since a large size (e.g., 16000^2) of matrix will cost a of computational time. Parallel image deconvolution algorithms would be useful for dealing with this problem.

The rest of the paper is organized as follows. In Section 2, we formulate the CDBS deconvolution problem. In Section 3, we describe our deconvolution algorithm. In Section 4, computations results on the PC Cluster are given to illustrate the usefulness of the parallel image deconvolution algorithm.

* Research supported in part by RGC Grant No. HKU 7132/00P and Science Faculty Collaborative Seed Grant, 2001.

2. FORMULATING THE CDBS DECONVOLUTION PROBLEM

The blurring of a spectrum is usually caused by the non-ideal characteristics of the recording component of the spectrometer. The observed spectroscopic signal at a given frequency contains contributions from the mathematical description of other frequencies due to instrument response. This phenomenon is known as convolution.

The actual spectral distribution $f(E_1,E_2)$ is smeared into the convoluted $g(E_1,E_2)$ by a resolution function $h(E_1,E_2)$ and an unknown random noise $n(E_1,E_2)$ in terms of the equation [4]:

$$g(E_1,E_2) = \int_{-\infty}^{\infty} h(E_1-E_2',E_2-E_2')f(E_1',E_2')\,dE_1'E_2' + n(E_1,E_2) \quad (1)$$

where $g(E_1,E_2)$ represents the intensity of the gamma ray recorded at several energy levels. The function $h(E_1,E_2)$ is called the instrumental profile of the spectrometer or impulse response or point spread function. A Ge detector scanning a monochromatic wave train of energy (E_x,E_y) in different directions records it so that it appears to contain an infinite range of wavenumbers. A spectrum which is really represented by

$$g(E_1,E_2) = \delta(E_1\text{-}E_x,E_2\text{-}E_y)$$

is recorded incorrectly by the instrument as

$$g(E_1,E_2) = h(E_1\text{-}E_x,E_2\text{-}E_y).$$

In other words, the observed spectrum is the integral of the product of the real spectrum and the instrumental profile [4]. Knowledge of the instrumental function $h(E_1,E_2)$ is usually known from experimentally calibration or simulation.

For a discrete spectrum, (1) can be expressed in the following equations in terms of matrices and vectors $\boldsymbol{g}$, $\boldsymbol{H}$, $\boldsymbol{f}$ and $\mathbf{n}$:

$$g_{i,j} = \sum_{k=1}^{n}\sum_{l=1}^{m} h_{ik,jl}f_{kl} + n_{ij} \quad (2)$$

where $i = 1,2,\cdots,n$ and $j = 1,2,\cdots,m$. In matrix form

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{n}. \quad (3)$$

3. DECONVOLUTION

The aim of the present work is to resolve the true spectrum obtained from CDBS image and to show that in principle a resolution approaching 1 mrad (ACAR equivalent) can be achieved by using suitable deconvolution algorithms.

Direct inversion by Fourier transform of equation (3) is the simplest approach to the deconvolution problem. The Fourier transform of the deconvoluted spectrum is obtained by "dividing" the Fourier transform of the known convolution (noise included) and the Fourier transform of the instrumental response function. Here the division refers to the multiplication by the inverse of the transformed matrix H. The deconvoluted spectrum can thus be obtained from the inverse Fourier transform. Nevertheless, the instrumental function $\boldsymbol{H}$ is often singular and if not, the inversion of $\boldsymbol{H}$ invariably gives rise to noise amplification at high frequencies [1]. Small errors (initial data errors or computer roundoff errors) in the measurement of the observed signal normally lead to large differences in the numerical solutions. Thus the solution is known as "ill-conditioned". The solution to this problem was given by Stokes [7], whose approach was that of rejecting the high frequency terms (i.e., introducing low pass filter), that corresponded to spectrum noise. The sharp cut in frequency space, however, invariably leads to high degree of "ringing" in the deconvoluted spectrum.

An alternative to the inversion of an ill-conditioned matrix is the generalized least squares method. Tikhonov [8] postulated an equation

$$J(\boldsymbol{f}) = \min_{\boldsymbol{f}} \{\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|_2^2 + \alpha\|\boldsymbol{f}\|_k^2\}. \quad (4)$$

The minimization problem is controlled by the choice of the regularization parameter α. Here $|\boldsymbol{f}|_k$ provides a measure of the size of $\boldsymbol{f}$ (when $k = 0$) or the kth derivatives of $\boldsymbol{f}$ (when $k > 0$) depending upon the particular choice of the norm $|\cdot|_k$. Large values of α yield solutions $\boldsymbol{f}$ with high smoothness but low sharpness of resolution while small values of α yield the solution $\boldsymbol{f}$ with low smoothness but high sharpness of resolution.

Moreover, both the calculated spectrum $\boldsymbol{f}$ and the observed spectrum $\boldsymbol{g}$ should not contain negative elements. This is imposing known a priori knowledge of $\boldsymbol{f}$ on the solution. The constraint on non-negative elements in the minimization problem is essential and turns out to be highly effective as a means of regularization. With non-negativity included minimization then may be written as:

$$J(\boldsymbol{f}) = \min_{\boldsymbol{f}\ge 0} \{\|\boldsymbol{g}-\boldsymbol{H}\boldsymbol{f}\|_2^2 + \alpha\|\boldsymbol{f}\|_k^2\}. \quad (5)$$

In the optimization with non-negativity constraint, we have to modify the Hessian which is dependent on the current solution $\boldsymbol{f}$ (Vogel [9]):

$$\boldsymbol{T} = \boldsymbol{D}_1 + \boldsymbol{D}_2\boldsymbol{H}\boldsymbol{D}_2 \quad (6)$$

where $\boldsymbol{D}_1$ is a diagonal matrix whose i-th diagonal entry is 1 if the i-th entry of

$$z(\boldsymbol{f}) = (\boldsymbol{H}^T\boldsymbol{H} + \alpha\boldsymbol{L})\boldsymbol{f} - \boldsymbol{H}^T\boldsymbol{g}$$

are non-negative, and is 0 otherwise. Similarly, D_2 is a diagonal matrix whose i-th diagonal entry is 0 if the i-th entry of $z(\boldsymbol{f})$ is non-negative, and is 1 otherwise. The iterative steps for the projected Newton method is as follows:

1. Set $\boldsymbol{f}^{(0)} = \boldsymbol{g}$
2. For $k = 0,1,2,\ldots$ until convergence
   (a) Compute the vector $z(\boldsymbol{f}^{(k)})$
   (b) Compute the matrices $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ which depends on $z(\boldsymbol{f}^{(k)})$
   (c) Set the matrix $\boldsymbol{T}$
   (d) Solve $\boldsymbol{T}\boldsymbol{p}^{(k)} = -z(\boldsymbol{f}^{(k)})$ by the conjugate gradient algorithm
   (e) Set $\boldsymbol{f}^{(k+1)} = \boldsymbol{f}^{(k)} + \beta\boldsymbol{p}^{(k)}$ where β is a line search parameter.

4. COMPUTATIONAL EXPERIMENTS

4.1 Monte-Carlo Simulation

The Monte-Carlo technique aims at mimicking the response of a CDBS spectrometer to gamma photons which are detected by the two back-to-back germanium detectors from a given material. To register a single event in the synthetic CDBS spectra, seven variates are required in the Monte-Carlo program. The first variate decides whether the positron is to annihilate with a core or valence (conduction band) electron. The annihilation gamma energy $E_{\gamma 1}$ is thrown according to a Gaussian distribution of standard deviation (modeling a metal). The second gamma energy $E_{\gamma 2}$ is constrained by energy conservation to $2mc^2\text{-}E_{\gamma 1}$. Resolution broadening associated with detection and electronics is then added to $E_{\gamma 1}$ and $E_{\gamma 2}$. The broadening is taken as pure Gaussian with

standard deviation = σ producing observed energies $E'_{\gamma 1}$ and $E'_{\gamma 2}$. Finally the action is effected by binding $E''_{\gamma 1}$ and $E''_{\gamma 2}$ into channels. Monte-Carlo spectra were thrown from 10^4 to 10^8 counts.

4.2 Deconvolution Results

Figures 1 and 2 shows the action of the Monte-Carlo CDBS program skipping resolution broadening in the procedures. The idealized linear momentum spectrum is shown in Figures 1 and 2. (Because the number of pages is limited, all the figures of this paper can be found in http://hkumath.hku.hk/~mng/physics. The same spectrum plotted on E_1 and E_2 histogram image is shown in Figures 1 and 2, but is only seen as a faint narrow diagonal line along $E_1 + E_2 = 1022$. The effect of "switching in" resolution broadening may be seen in Figures 3 and 4. Figures 5 and 6 shows the instrumental resolution function corresponding to Figures 3 and 4. An example is shown in Figure 7 and 8 where synthetic data for Li has been deconvoluted. The deconvoluted image is seen to lie much nearer to the line $E_1 + E_2 = 512$ in a narrow band and there are no signs of any negative ripples. Figure 9 shows a cross-section across the $E_1 - E_2$ momentum direction, and while showing the presence of artificial ripples at high momentum confirms the non-negativity of the deconvoluted image. In general, it may be said that the imposing of the a priori knowledge of non-negativity has greatly improved the quality of the deconvolution. For comparison, the deconvoluted image without the non-negativity constraint is shown in Figure 10. The result is demoralizing as some negative components appear in the spectra.

4.3 Parallel Computation Results

In this subsection, we describe the performance of the algorithm using the fast Fourier transform on the PC Cluster in The University of Hong Kong.

With the continuous reduction in cost and increase in processing power of PCs, there has been a notable development in the world to set up large clusters of PCs to complement their installations of the traditional packaged supercomputers. These systems are generally called Beowulf Clusters because this design was originally started by the Beowulf project at NASA Goddard Space Flight Center in the U.S. The Computer Centre, The University of Hong Kong has acquired 33 nodes of Compaq Proliant system for our users to use this kind of computer system. Beowulf clusters could be an alternative solution to augment our existing high performance computing facility based on our IBM SP2 parallel computer. There are 32 computation nodes. They are used for batch processing. Each node is comprised of Compaq Proliant server and consists of Compaq ProLiant DL360 Rackmount server, dual PIII 1GHz with 133MHz GTL bus, 256KB level two ECC cache, 2GB RAM, integrated Dual Wide-Ultra2 controller, 18.2 GB Pluggable Wide Ultra SCSI-3 drive, 24XCDROM, 1.44 FDD, two Compaq 10/100 T/X Embedded PCI Intel UTP and integrated RAID standard In addition, the master node contains 200GB of RAID 5 disk. The network is the "3Com SuperStack III switch 4300 48 port fast ethernet switch" links up all computation nodes and the master nodes. The master node is also connect to the campus network using the second ethernet adaptor.

All tests used double precision complex floating point computations. Communication was performed using MPI.

In Table 1, we show the percentage of time spent by two parallel programming parts: two-dimensional fast Fourier transforms (FFTs) and the other parts including the Newton/CG method. We see that most of the processing time is spent in the computation of two-dimensional FFTs. The speedup of two-dimensional FFTs is an important factor for an efficient implementation of this image deconvolution algorithm. The typically larger speedups observed when the size of the FFT is large stem from the larger amount of arithmetic for the computation of the FFT. We attribute this to the workload between the communication among processors and the amount of computation required by the FFT, see Table 2. In Table 3, we show the total processing time required for different sizes of the problem and different number of processors. We see that our implemented parallel image deconvolution is quite scalable especially when the size of the problem is large, see Table 4. (The scale-up is defined as the total processing time required by k processors (k = 4,8,16) over the total processing time required by 2 processors.)

5. CONCLUDING REMARKS

The image processing algorithm presented here enable efficient parallel solution of the image deconvolution problem. We have tested our method on the PC Cluster. The method displays high efficiencies in an implementation on a parallel computer.

Table 1 Percentage of time spent by different parallel programming

Size	Number of processors	FFTs	Others
512	2	58.7%	41.3%
512	4	65.1%	34.9%
512	8	76.6%	23.4%
512	16	76.8%	23.2%
1024	2	59.9%	30.1%
1024	4	65.2%	34.8%
1024	8	74.9%	25.1%
1024	16	68.3%	31.7%
2048	2	60.0%	40.0%
2048	4	62.1%	37.9%
2048	8	67.6%	32.4%
2048	16	69.8%	30.2%

Table 2 Percentage of time spent by FFTs using 16 processors

Size	Computation	Communication
512	18.0%	82.0%
1024	35.4%	64.6%
2048	34.0%	66.0%
4096	36.7%	63.3%
8192	40.7%	59.3%

Table 3 Total processing time

Size	Number of processors			
	2	4	8	16
512	115.7	65.2	45.7	40.2
1024	467.4	250.5	158.7	73.4
2048	1893.4	1108.9	584.7	302.3
4096	–	4706.2	2437.8	1281.3
8192	–	–	–	5405.9

Table 4 Scale-up

Size	Number of processors		
	4	8	16
512	1.77	2.53	2.88
1024	1.87	2.95	6.37
2048	1.71	3.24	6.26

6. REFERENCES

[1] E. Angel and A. Jain.Applied Optics, 17-4, 2186 (1978).

[2] C. Beling, M. Li, Y. Shan, S Cheung, S. Fung, B Panda and A. Seitsonen, J. Phys: Condens. Matter, 10, 10475 (1998).

[3] D. Britton, W. Junker and P. Sperr. Materials Science Forum, 105, 1845 (1992).

[4] J. Jansson. Deconvoluting of Images and Spectra, Academic Press, 1984.

[5] L. Liszkay, C. Corbel, L. Baroux, P. Hautojarvi, M. Bayhan, A. Brinkman, S. Taraenko, Applied Physics Letter 64, 380 (1994).

[6] R. MacDonakd, K. Lynn, R. Boie and M. Robbins. Nuclear Instr. Meth. 153, 189 (1977).

[7] A. Stokes. Proc. Phys. Soc., 61, 382 (1948).

[8] A. Tikhonov and V. Arsenine. Methodes de resolution de problemes mal poses (Mir Moscow, 1976).

[9] C. Vogel. Scientific Computing, Springer, Singapore, 148 (1997).

Numerical Method for Extreme Wind Synthesizing

Edmond D. Cheng
Department of Civil and Environmental Engineering
University of Hawai'i at Manoa, Honolulu, Hawaii 96822, U.S.A.
E-mail: **edcheng@hawaii.edu**

ABSTRACT

In order to provide a realistic basis of determining design extreme wind speeds on structures, an extreme wind generation model is presented herein. This computational procedure is a stochastic model of generating long-term annual extreme winds on the basis of short period of records. Basically, this method uses historical wind data to establish Markov transition probabilities at an intended project site. These probabilities will be the guide for producing synthesized wind speeds of a desired duration. The simulation model consists of four interactive sub-programs and numerous sub-routines. A parallel operational rules were established and an expert system shell is utilized to facilitate the application of the proposed model.

Keywords: Extreme Wind, Markov Stochastic Model, Knowledge-based Expert System.

1. INTRODUCTION

The probability of structural failure could be estimated following the rules set forth by the principles of structural reliability provided the following are given: the acceptable probability of failure, the probability distribution of the wind loads, and the load resisting capabilities of the members and connections of the structure. For structures of some complexity, this type of information is difficult to evaluate. Therefore, the structural reliability analysis in the present state-of-the-art is incapable of establishing a set of general criteria of structural failures under such dynamic loading as extreme winds [1, 2, 3].

The basic strategy underlying this proposed Markov stochastic model is to have the time series of hourly wind speeds generated in parts: for those winds associated with well-behaved climates, in which extraordinary strong winds will not be expected to occur, and for those extreme winds belonging to tropical cyclones or some other non-tropical cyclones. In other words, consider these to be generated time series of hourly wind speeds as a time-dependent system, and this system is decomposed into a set of components for which parallel operating rules of these components may be designed. Under these parallel operational rules, a

system of wind data will be constructed. Recently, we realized that the application of the proposed simulation model, for inexperienced users, is tedious and painstaking, therefore, an expert system shell is utilized to facilitate the application of the simulation model. An application of this stochastic model is demonstrated.

2. SIMULATION MODELS

Two major assumptions are made in this paper: (1) the correlation between two adjacent hourly wind speeds depends only on the time interval between them, and (2) the degree of expected persistence between successive wind speeds does not depend on the level of the wind speeds. The following steps are necessary to construct the model.

2.1 State Divisions

The first step is to divide the entire range of observed speeds into a finite member of states. This task shall be performed with reference to the probability histogram, which is derived from the observed wind data at a site. The state divisions should be made in principle of even distribution of measured wind speed values among the states; therefore, state intervals will result in various non-equal sub-ranges of wind speeds.

2.2 Distribution Functions

The second step in the model is the probability density functions (PDF) and the cumulative distribution functions (CDF) of wind speeds in various states. In this paper, three types of PDF are utilized to fit a wind speed histogram, viz., uniform, linear, and Rayleigh functions. If relative uniform wind speeds were observed within a state or states, constant values were assigned to the state of states. Likewise, if linear wind speed variations were observed in a state, a linear PDF is assigned to that state. The Rayleigh PDF is exclusively reserved for the last state in order to take care of extreme winds.

2.3 Transition Probability Matrices

In transition probability, p_{ij} is defined as the probability of a wind speed in state j which will occur in the next hour, given that a wind speed in state i has occurred in this hour.

For a wind field of m finite states, p_{ij} is actually a conditional transition probability of wind speed V_t going from state i at hour t to wind speed V_{t+1} going from state i at hour t to wind speed V_{t+1} of state j at hour t+1, or

$$P_{ij} = P(V_{t+1} = j | V_t = i) \qquad (1)$$

with m states determined, an mxm transition probability matrix PM, can be determined as

$PM = [P_{ij}]$ for $i,j = 1,2,\ldots,m$.

in which, p_{ij} have the following properties:

$$\sum_{j=1}^{m} p_{ij} = 1, \qquad \text{for } i = 1,2,\ldots,m.$$

and

$p_{ij} \geq 0$, for all i and j.

In the process of determining a transition probability matrix, first, an mxm tally matrix, *TM*, is computed from historical records as

$TM = [f_{ij}]$, for $i,j = 1,2,\ldots,m$.

where f_{ij} = the number of transitions of V_t going from state i at hour t to V_{t+1} of state j at the next hour within a time period under consideration. Then the transition probability, p_{ij}, can be estimated from the tally matrix as follows:

$$p_{ij} = f_{ij} / \sum_{j=1}^{m} f_{ij} \qquad \text{for } i,j=1,2,...,m. \qquad (2)$$

In this paper, a day was divided into several diurnal periods. Similarly, the variation of mean monthly wind speeds are accounted for by grouping consecutive months with similar wind speed trends into a number of seasons for a year. Let the number of periods in a day and the number of seasons in a year be R and S, respectively, then there will be RxS transition probability matrices in the simulation process.

A typical tally matrix for a given period r and season s can be expressed as

$$TM(s,r) = [f_{ij}^{s,r}]$$

Therefore, by means of Equation (2), the transition probability matrix will be

$$PM(s,r) = [p_{ij}^{s,r}] \quad (3)$$

where s = 1,2,...,S, and r = 1,2,...,R.

3. EXSYS-PROFESSIONAL

We realized that the application of our simulation model, by inexperienced users, is tedious, painstaking and time-consuming. Therefore, our current effort is to utilize an expert system shell to facilitate the application of the simulation model in a microcomputer environment.

The chosen system is a rule-based system called EXSYS-Professional [4] which is coded in C language. Although this system, like many other PC tools, represents knowledge as IF-THEN rules and uses backward chaining to process these rules, it is capable of executing external programs and can also pass its results to external programs. Furthermore, the EXSYS-Professional has a convenient command language. This language is utilized to control the direction and flow of our knowledge-based expert system (KBES), while the rules provide the logic, inference and chaining capabilities. The EXSYS-Professional was selected because of these powerful features.

4. APPLICATION

A schematic block diagram of the KBES simulation model is illustrated in Figure 1. This model is applied to wind data collected at a station in Texas. Studies on wind patterns in Texas have shown that daily periodical phenomenon as well as a strong seasonal persistence was clearly identified. For illustration, historical hourly wind data (1 January 1981 to 31 December 1992) at Lubbock, Texas were utilized in the simulation model.

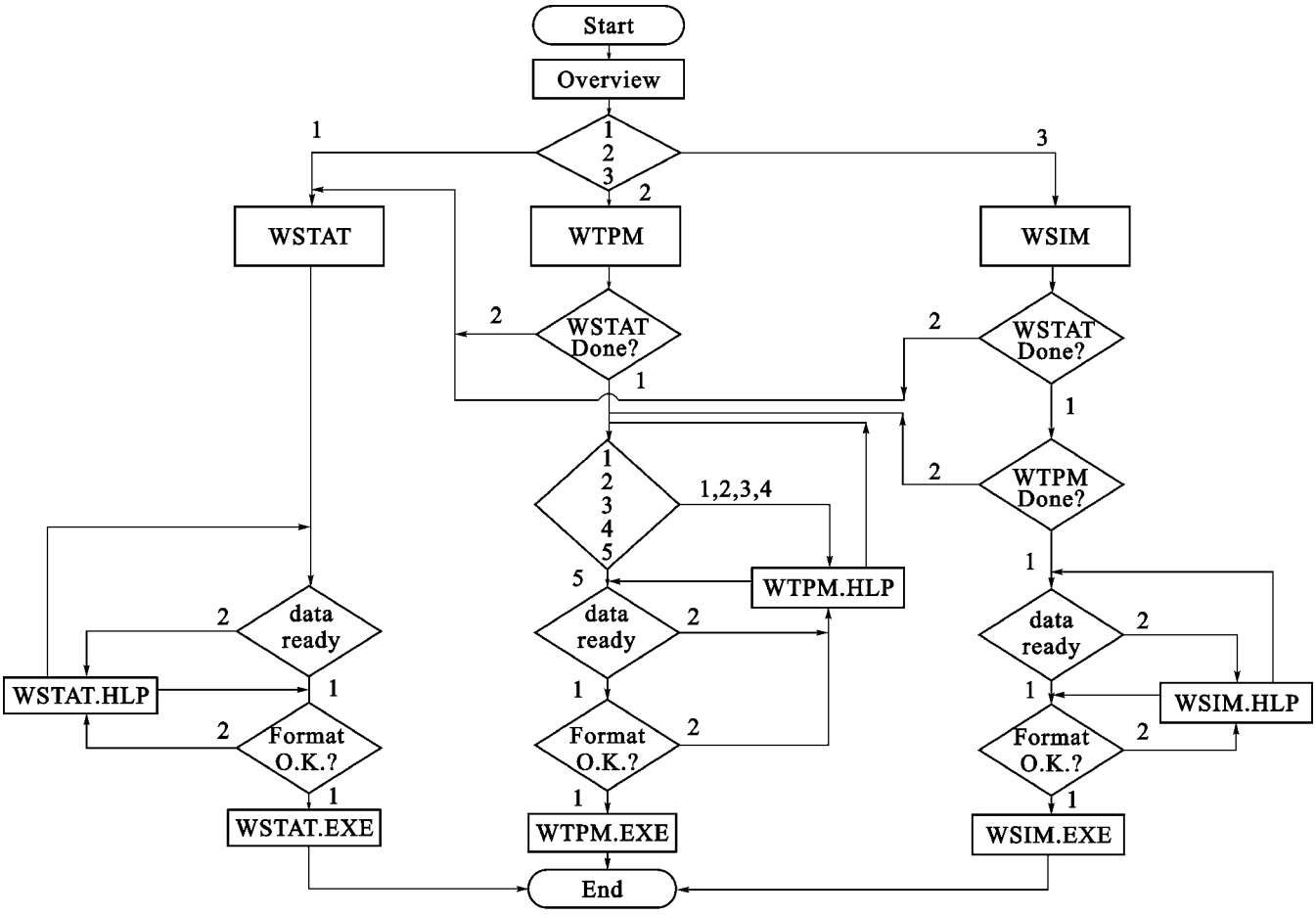


Figure 1 A Schematic Block Diagram of the Knowledge-Based Expert System Simulation Model

In the process of simulation, nine state intervals of hourly wind speeds, two periods in a day (8 p.m. - 9 a.m. and 10 a.m. - 7 p.m.), and three seasons in a year (February -May, June-October and November-January) were considered.

Both tests of Markov property and stationarity of the 12 years' wind records were passed.

Based on the 12-year (1 January 1981 – 31 December 1992) hourly wind records available at Lubbock, Texas, three

computer simulation runs were made. Each run generated 100-year hourly wind data. Data bases for the three runs are: the first 5-year (January 1981 - December 1985); the next 5-year (January 1986 - December 1990) and all the 12-year. The annual extremes of the simulated and the 41-year (1950-1990) historical data at Lubbock are fitted with Type I distribution. Using the Gumbel line of the historical data as the reference, the performance of the proposed simulation model is summarized in Table 1.

Table 1 Estimated extreme winds from historical records and from simulation method at Lubbock, Texas

Data Period	Recurrence Interval (yr)	V_A	S_V	V_S	$(V_A-V_S)/S_V$
1950-1990	25	28.4	1.14		
	50	30.0	1.36		
	100	31.6	1.59		
1/81-12/85	25			29.1	-0.61
	50			30.1	-0.07
	100			31.0	0.38
1/86-12/90	25			29.1	-0.61
	50			30.0	0
	100			30.9	0.44
1/81-12/92	25			29.3	-0.79
	50			30.3	-0.22
	100			31.3	0.19

V_A=fastest-mile wind speed from annual series in m/s and 10 m above ground level
V_S= simulated fastest-mile wind speed in m/s and 10 m above ground level
S_v=Cramer-Rao's standard deviation of inherent sampling error of historical records.

In this table, the deviations of the simulated 25-year, 50- year or 100-year wind speeds from the reference Gumbel line were measured by S_v, Cramer-Rao's standard deviation of the inherent sampling error of the historical records [5]. As indicated in Table 1, the differences between the simulated annual extreme wind speeds and the values obtained from the referenced Gumbel lines of the historical data at 25-year or 50-year or 100-year recurrence intervals are almost all smaller than one S_v. This result is very encouraging.

5. CONCLUSIONS

A procedure for predicting design extreme winds up to 100 years recurrence interval is demonstrated. Although these extreme winds were predicted on the basis of data collected over a rather short period of time, the results obtained from the application of this simulation model may be considered reasonable.

6. REFERENCES

[1] E.D. Cheng, "Wind Data Generator: A Knowledge-based Expert System," Journal of Wind Engineering and Industrial Aerodynamics, Vol. 38, 1991, pp. 101-108.
[2] E.D. Cheng, "Generalized Extreme Gust Wind Speeds Distributions," Journal of Wind Engineers, Vol. 89, 2001, pp. 653-656.
[3] E.D. Cheng, A.N.L. Chiu, "Short-record-based Extreme wind Simulation," Journal of the U.S. National Institute of Standards and Technology, Vol. 99, No. 4, 1994, pp. 391-397.
[4] EXSYS-Professional, New commands and features. EXSYS, Inc., P. O. Box 11247, Albuquerque, NM 87192, U. S. A., 1989.
[5] E. J. Gumbel, Statistics of Extremes. New York: Columbia University Press, 1958.

A Distributed QoS Multicast Routing Algorithm*

Li Layuan Li Chunlin
Department of Computer Science. Wuhan University of Technology. Wuhan 430063.P.R.China
E-mail: jwtu@public.wh.hb.cn

ABSTRACT

With the rapid development of Internet, mobile networks and high-performance networking technology, QoS multicast routing has become a very important research issue in the areas of networks and distributed systems. This is also a challenging and hard problem for the next generation Internet and high-performance networks. It attracts the interests of many people. This paper presents a distributed QoS multicast routing algorithm (DQMRA). This algorithm deals with delay and bandwidth constraints, and has low cost. The DQMRA attempts to significantly reduce the overhead for constructing a multicast tree. In this paper, the proof of correctness of the DQMRA is given, and the performance of the algorithm is evaluated using simulations. The study shows that our algorithm provides an available approach to QoS multicast routing.

Keywords: QoS; multicast; QoS constraints; routing algorithm

1. IMPORTANT INFORMATION

Many simple routing algorithms, which were mainly concerned with connectivity, were developed based on the network topology. The situation is, however, different now with emerging real-time multimedia applications like digital audio and video. These applications express their quality of service (QoS) requirements in terms of bound on end-to-end delay, packet loss probability etc. Routing algorithm, while making a routing decision, should consider the QoS needs of an application in order to make an efficient use of network resources.

In recent years, there has been an effort, at both the algorithm and protocol levels[1~21],to develop multicast mechanisms which meet the QoS need of the multicast services. These services have been used by various continuous media applications. For example, the multicast back-bone (Mbone) of the Internet has been used to transport real time audio/video for news, entertainment, video conferencing and distance learning. The main goal in developing multicast routing algorithm is to minimize the communication resources used by the multicast session. This is achieved by minimizing the cost of the multicast tree, which is the sum of the costs of the edges in the multicast tree. The simplest approach of finding a multicast tree is to find the shortest paths from the source to each destination separately and then merge the resulting paths to form a tree. PIM-DM [6] is an example of a protocol that uses this heuristic to route multicast sessions. The best known heuristics were proposed by Kou et al. (KMB heuristic)[8], Takahashi and Matsuyama (TM heuristic)[6], and Rayward-Smith (RS heuristic)[6].PIM-SM[4] and CBT[3] are examples of routing protocols that use shared tree approach. It has been shown that the cost of the tree generated by these heuristics are at most twice as much as the optimum tree. In contrast, the ratio of the cost of the tree generated by shortest path heuristic (SPH) [8] and the cost of the optimum tree can be as large as the number of destinations.

The above algorithms' limitation is that they only consider delay constraint. It is difficult to extend them to handle multiple constraints. QoSMIC[6], proposed by Faloutsos et al., alleviates but does not eliminate the flooding behavior. In addition, an extra control element, called Manager router, is introduced to handle the join requests of new members. This paper presents a new heuristic algorithm for the QoS multicast routing (DQMRA). DQMRA mainly deals with delay and bandwidth constraints, which attempts to significantly reduce the overhead for constructing a multicast tree.

The rest of the paper is organized as follows. Section 2 describes some definitions. Section 3 presents the DQMRA. Section 4 gives the correctness proof. Some simulation results are provided in Section 5. The paper concludes with Section 6.

2. SOME DEFINITIONS

Let $G=(V,E)$ be a graph, where V denotes a set of vertices and E denotes a set of symmetric links. Each link (u,v) is assigned a cost $C(u,v)$ and a delay $De(u,v)$.The link delay represents the waiting and transmission delay incurred by a packet at a given node. Furthermore, let $s \in V$ be a source node and $D \subset V$ be a set of destination nodes. Every node $d \in D$ is associated with an end-to-end delay bound Δ_d. Let R be the positive weight and R^+ be the nonnegative weight. For any Link $e \in E$,we can define the some QoS metrics: delay function *delay* (e):$E \rightarrow R$, and delay jitter function *delay-jitter* (*e*):$E \rightarrow R^+$. Similarly, for any node $n \in V$,one can also define some metrics: delay function delay (*n*):$V \rightarrow R$, cost function cost (*n*):$V \rightarrow R$, delay jitter function *delay-jitter* (*n*): $V \rightarrow R^+$ and packet loss function *packet-loss* (*n*):$V \rightarrow R^+$. We also use $T(s,M)$ to denote a multicast tree, which has the following relations:

1) $delay(p(s,t)) = \sum_{e \in P(s,t)} delay(e) + \sum_{n \in P(s,t)} delay(n)$.

2) $cost(T(s,M)) = \sum_{e \in T(s,M)} cost(e) + \sum_{n \in T(s,M)} cost(n)$.

3) $bandwidth(p(s,t)) = \min\{bandwidth(e), e \in P(s,t)\}$.

4) $dealy\text{-}jitter(p(s,t)) = \sum_{e \in P(s,t)} delay - jitter(e) + \sum_{n \in P(s,t)} delay - jitter(n)$.

5) $packet\text{-}loss(p(s,t)) = 1 - \prod_{n \in P(s,t)} (1 - packet\text{-}loss(n))$

where $p(s,t)$ denotes the path from source s to end node t of $T(s, M)$.

QoS-based multicast routing problem mainly deals with some elements: Network $G=(V,E)$, multicast source $s \in V$, the set of end nodes $M \subseteq \{V-\{s\}\}$, $delay(\cdot) \in R$, $delay\text{-}jitter(\cdot) \in R^+$, $cost(\cdot) \in R$, $bandwidth(\cdot) \in R$, and $packet\text{-}loss(\cdot) \in R^+$. This routing problem is to find the $T(s, M)$ which satisfies some QoS

* The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

constraints:

1) Delay constraint: *delay* (p (s,t))$\leqslant De$

2) Bandwidth constraint: *bandwidth* (p (s,t))$\geqslant B$

3) Delay jitter constraint: *delay-jitter* (p (s,t))$\leqslant J$

4) Packet loss constraint: *packet-loss* (p (s,t))$\leqslant L$

Meanwhile, the *cost* (T (s, M) should be minimum. Where De is delay constraint, B is bandwidth constraint, J is delay jitter constraint and L is packet loss constraint. In the above QoS constraints, the bandwidth is concave metric, the delay and delay jitter are additive metrics, and the packet loss is multiplicative metric. In these metrics, the multiplicative metric can be converted to the additive metric. For simplicity, we assume that all nodes have enough resource, i.e., they can satisfy the above QoS constraints. Therefore, we only consider the links' or edges' QoS constraints, because the links and the nodes have equifinality to the routing issue in question. In this paper, we mainly consider two QoS constraints: delay and bandwidth. The characteristics of edge can be described by a three-tuple (De,B,C), where De,B and C denote delay, bandwidth and cost, respectively.

3. DQMRA DESCRIPTION

The basic idea of the is to first build a minimal cost tree using a SPH-based approach DQMRA. The link cost, computed based on fair-share load cost function, takes into consideration both the load of currently supported connections and the load of the new connection. Using this minimal cost tree, delay bounds are checked and violations are removed accordingly.

The basic steps of the algorithm can be described as follows:

1) Remove the link that violates bandwidth constraint.

2) Initially, set i =1. Construct a subtree, T_1= (s,$\varnothing$), consisting of the source node s.

3) Set i=i+1. Find the closest node, d_i, to T_{i-1} such that $d_i \in (D-D_{i-1})$ (ties are broken arbitrarily). Construct a new subtree, T_i, by adding all edges and nodes in PATH (d_i, T_{i-1}) to T_{i-1}. Set D_i=$D_{i-1} \bigcup \{d_i\}$.

4) If $|D_i|<|Z|$ then go to step 2); otherwise, done (T_i is the final tree).

5) Find the set of nodes W whose path delay, $P \forall d \in D$, is larger than $\triangle_d$ ($P_\delta(s,d) > \triangle_d$) and remove them and their relay-paths from T.

6) For each node $v \in W$, find the shortest delay path from s to v and add that path to the tree. If the new path causes a node to have two incoming paths, then remove the relay part of the path whose delay is larger.

Where, T_i be a subtree, $d_i \in D$ be a multicast node, *PATH* (d_i, T_i) be the shortest path from d_i to T_i, and Z denotes the number of a group members. The above algorithm is similar to Prim's Minimum Spanning Tree algorithm. The time complexity of building the SPH tree is $O(n^2 log(n))$, where n is the number of nodes in the network. However, the need to compute the shortest path at each stage raises the total complexity of the algorithm to $O(|Z|\cdot n^2)$. It was shown by simulation that on average the additional cost of the trees produced by this algorithm is no more than 5% above the cost of the corresponding optimal Steiner trees. However, in the worst case, the cost of this algorithm is $(2-2/|Z|)\times C(T_{opt})$, where $C(T_{opt})$ is the cost of the corresponding optimal Steiner tree.

After building a low cost multicast tree, DQMRA checks the delays from the source to all destinations. The path to any node, whose delay is higher than the delay bound, is replaced with the shortest delay path from the source to that node. That is achieved by removing the violating node and the relay-path to that node. A *relay-path* is the path in which all nodes are of degree two and are not multicast nodes, excluding the end nodes.

Because the above heuristic uses the shortest delay paths, it finds a solution if one exists. Furthermore, if there is no violation, or the number of violations is low, then the cost of the resulting tree is close to the cost of the optimal tree. The complexity of the algorithm is bounded by the complexity of SPH, which is $O(|Z|\cdot n^2)$. Fig.1 is an example network, where node 1 is the source and nodes 2,3,4, and 8 are the destinations. Node 8 has a delay of 11, which violates the delay bound (suppose delay bound be 10). The algorithm uses the least delay path (1,4,8) to connect node 8 as shown in Figure 2 (see the bold lines).

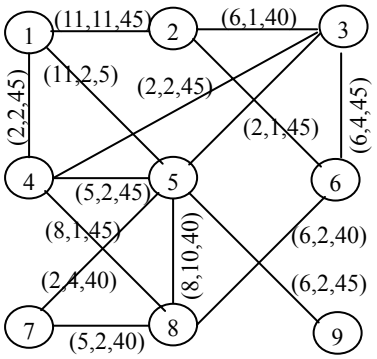


Fig.1 An example network

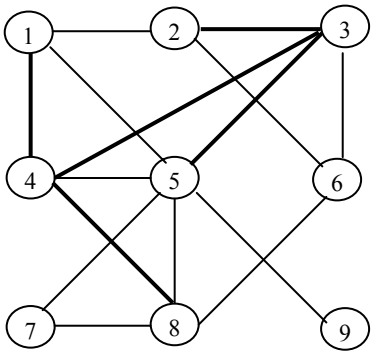


Fig.2 Steiner minimum tree

If the path fails to satisfy the delay bound, the minimum cost path (MCP) algorithm is used to find constrained path. MCP algorithm can be described as follows.

1) Create a new graph G' consisting of G's nodes and G's edges reversed.

2) Create, P, the set of the shortest delay paths from v to T in G'.

3) Remove the edges of T from G'.

4) Find the set of the least cost paths from v to T in G' and add them to P.

5) Out of P, select the path p such that delay (p) $<\triangle v$ and cost (p) is minimum.

6) If no such path exists then return "no solution" else return p.

MCP searches for the least-cost, delay-bounded path that connects the delay violating node, $v \in D$, to the current tree, T. This is achieved by building a new graph G', obtained by reversing the links of G while keeping all its nodes. The shortest delay paths from v to the tree nodes are computed first. Then, the shortest cost paths from v to T's nodes are also computed. But prior to computing the shortest cost paths. T's links are removed from G' to create different in dependent least cost paths. In other words, without removing T's links, it

is very likely that the shortest paths use *T*'s links and that will lead to creating paths that are derived from each other.

4. THE CORRECTNESS

Theorem 1. In MCP algorithm, the routing is cycle free.
Proof. Using MCP algorithm to find the least constrained path in the above heuristic, when there is a delay bound violation, may create a cycle or a node with more than one incoming link. Since the source node never has an incoming path, a cycle can only be created by adding another incoming path to one of the other tree nodes. Therefore, in any cycle there must be at least one node with two incoming paths. As was explained, this type of problem is dealt with by removing the relay-paths whose delay values are higher. Hence, the theorem holds.
Theorem 2. If a feasible path that satisfies the delay and bandwidth metrics exists, then it must be searched by DQMRA.
Proof. If a solution that satisfies the delay bound exists then there exists a path from the source to each destination, $v \in D$, such that the delay on that path is less than $\triangle v$. If any delay violation occurs the above heuristic replaces the delay violating path by the least delay path from the source to the delay-bound violating destination. Since such a path exists, the replacement procedure leads to a solution which satisfies the delay and bandwidth constraints of the multicast nodes. Thus, the theorem holds.
Theorem 3. If DQMRA terminates without an available and feasible path, all nodes out of $T(s,M)$ are ether initial state or in failure state.
Proof. DQMRA terminates without success only when the new member's join request is rejected, i.e., it changes into the failure state. Since the new member is leaf node of the searching tree, when it changes into the failure state, all nodes in the searching tree must be in the failure state. The nodes outside the searching tree remain in the initial state.

5. SIMULATIONS

We used simulation for our experimental investigations to compare the performance of different routing algorithms. Based on Waxman's [6] algorithm, a random generator was used to generate networks. In this model, an edge is introduced between a pair of nodes with the following probability:

$$P(u,v)=\beta\left(-\frac{d(u,v)}{Da}\right)$$

where, $d(u,v)$ is the distance between two nodes u and v and D is the maximum distance between any two nodes in the network. The a and β are two parameters in range (o,1) that control the properties of the generated graph. Large value of β increases the degree of the node and small value of a results in the increase to density of shorter edges to the longer ones. The random graphs were generated using the above algorithm with an average degree 4 and a high density of shorter edges.
We generate random network of size 50 to resemble real networks. Each of the 50 nodes is distributed across a Cartesian coordinate plane with minimum and maximum coordinates (0,0) and (50, 50), creating the forest of 50 nodes spread across this plane. The nodes are then connected using the Waxman's algorithm in the following manner. The edges in the network are introduced by considering every possible node pair (u,v) and generating a random number $0 \leqslant p < 1$. If p is less than the value of probability function $P(u,v)$, that depends on the Euclidean distance between u and v, then an edge is introduced between the node pair. The generated network is then tested for connectivity and the process is repeated till a single connected network is produced. In our experiments, we use random networks with an average node degree of 4, which is close to the average node degree of the current Internet. Further, we make sure that the minimum node degree is 2 and maximum node degree is 10 in the generated random graphs.
The incoming traffic having a *token bucket filtered process* characteristic was considered with burst size, b=1 Kbyte, and long time average rate, 10Mb/s. The maximum length of the packet was set to 100 bytes. The link capacity (C) of all the links in the network was set to 200 Mb/s. We note here that since we are not taking into consideration propagation delay in this study, the delay bound primarily consists of queuing delay. A multicast session was rejected if the delay bound is not satisfied for any of the destinations in the multicast group or if the bottleneck bandwidth for any source destination pair did not meet the minimum bandwidth requirement of the application.

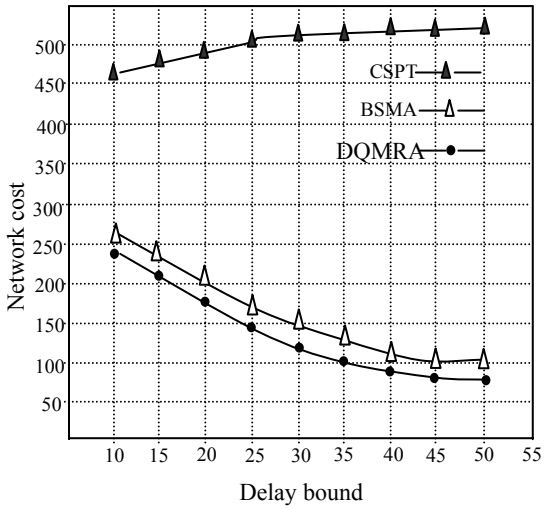


Fig.3 Network cost vs. delay bound

Fig.3 is the network cost versus delay bound. The D values on the X-axis are the mean the network cost of the CSPT [8] algorithm is on the top and almost does not change as D increases. This is because the generation of the shortest path tree does not depend on D. From Fig.3, one can also see that tree costs decrease for BSMA and DQMRA as the delay bound is relaxed. Of three algorithms, the proposed DQMRA has the lowest cost. Fig.4 shows the network cost versus group size. In this round of simulations, the network size is set to 300. From Fig.4, one can see when group size grows, the network cost produced by BSMA and DQMRA increases at a rate much lower than CSPT. DQMRA and BSMA can produce trees of comparable costs.

6. CONCLUSIONS

This paper has presented a heuristic algorithm for QoS multicast routing (DQMRA)DQMRA attempts to significantly reduce the overhead for constructing a multicast tree with delay and bandwidth constraints, and low cost. We formally verified its correctness. Some simulation results are given. Our study shows that DQMRA is an available and feasible

approach to QoS multicast routing.

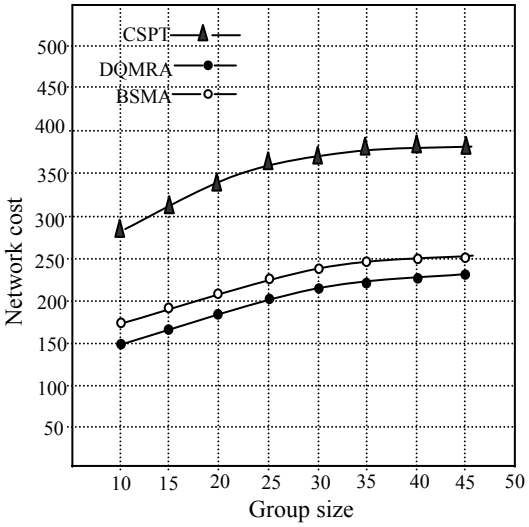


Fig.4 Network cost vs. group size

7. REFERENCES

[1] Li Layuan and Li ChunLin, "The QoS routing algorithm for ATM networks", *Computer Communications*, No.3-4, Vo1.24,2001, pp. 416-421.

[2] Li Layuan. "A formal specification technique for communication protocol". *Proc of IEEE INFOCOM*, April. 1989, pp. 74-81.

[3] K. Carberg and J. Crowcroft. "Building shared trees using a one-to-many joining mechanism". *ACM Computer Communication Review.* Jan. 1997, pp. 5-11.

[4] T. Ballardie, P. Francis and J. Crowcroft, "An architecture for scalable inter-domain multicast routing," *ACM SIGCOMM,* Sept. 1993, pp. 85-95.

[5] X. Jia. "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks', *IEEE/ACM Transactions on Networking,* No.6, Vo1.6, Dec. 1998, pp. 828-837.

[6] Li layuan and Li Chunlin, Computer Networking, National Defence Industry Press, Beijing, 2001.

[7] Moses Chrikar, Joseph Naor and Barch Schieber, Resoutce optimization in QoS multicast routing of real-time multimedia, *proc of IEEE INFOCOM*. 2000. pp. 1518-1527.

[8] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum- cost multicasting," *Proc. IEEE INFOCOM 95, Boston, MA,* April 1995.

[9] Li Layuan and Li Chunlin, "A multicast routing protocol with multiple QoS constraints." *Proc of WCC*, Aug. 2002.

[10] Zhang Q, Lenug Y W, "An orthogonal genetic algorithm for multimedia multicast routing." *IEEE Trans Evolutionary Computation,* 1999, 3(1): 53-62.

[11] F. Xiang, L. Junzhou. W. Jieyi and G. Guanqun. "QoS routing based on genetic algorithm", *Computer Communications,* 22(1999), pp. 1392-1399.

[12] Dean H. Lorenz and Ariel Orda. "QoS routing in networks with uncertain parameters" *IEEE/ACM Transactions on Networking.* Vol.6, No.6, DEC.1998, pp. 768-778.

[13] Li Layuan and Li Chunlin, "A routing protocol for dynamic and large computer networks with clustering topology," *Computer Communication,* No.2, Vol.23, 2000, pp. 171-176.

[14] D. G. Thaler and C. V. Ravishankar, "Distributed center-location algorithms," *IEEE JSAC,* Vol.15, April 1997, pp. 291-303.

[15] I. Cidon, R. Rom, and Y. Shavitt, "Multi-path routing combined with resource resetvation," *Proc of IEEE INFOCOM*, April 1997, pp. 92-100.

[16] J. Mog. "Multicast routing exeensions to OSPF." RFC 1584. March.1994.

[17] Y. Xiong and L.G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks,' *IEEE Trans on Networks*, Vol. 7, No. 1, Feb, 1999, pp. 98-110.

[18] S. Chen and K. Nahrstedt, "Distributed QoS routing in ad-hoc networks," *IEEE JSAC*, special issue on ad-hoc networks, Aug. 1999.

[19] Li Layuan. "The routing protocol for dynamic and large computer networks". Journal of computers, No.2, Vol.11, 1998, PP. 137-144.

[20] B. M. Waxman. "Routing of multipoint connections." *IEEE Journal of Selected Area in Communications*, Dec. 1998, pp. 1617-1622.

[21] R.G. Busacker and T. L. Saaty, Finite Graphs and Networks: An introduction with applications, McGraw-Hill, 1965.

LI Layuan was born in Hubei, China on 26 February 1946. He received the BE degree in Communication Engineering from Harbin Institute of Military Engineering, China in 1970 and the ME degree in Communication and Electrical Systems from Huazhong University of Science and Technology, China in 1982. He academically visited Massachusetts Institute of Technology (MIT), USA in 1985 and 1999, respectively. Since 1982, he has been with the Wuhan University of Technology (MUT), China, where he is currently a Professor and Ph.D. tutor of Computer Science, and Editor in Chief of the Journal of WUT. He is Director of International Society of High-Technol. and Paper Reviewer of IEEE INFOCOM, ICCC and ISRSDC. He was the head of the Technical Group of Shaanxi Lonan PO Box 72, Ministry of Electrical Industry, China from 1970-78.His research interests include high speed computer networks, protocol engineering and image processing. Professor Li has published over one hundred and fifty technical papers and is the author of six books. He also was awarded the National Special Prize by the Chinese Government in 1993.

LI Chunlin was born in Wuhan, China in 1974. She received the BE and ME degrees in Computer Science from Wuhan University of Technology, China in 1997 and 2000, respectively. She is currently a Ph. D. candidate in computer science of HUST. Her research interests include Internet/Intranet, distributed computing and multimedia technologies. She received Wu Fu award by Ministry of Communications, China in 1997.

Research and Implementation of Topology Discovery Algorithms in Campus Networks*

Liu Yu-Hua[1),2)] , Yang Jin-Gui[1)] and Xiao De-Bao[1)]
[1)] Department of Computer Science,Central China Normal University, Wuhan 430079 ,China
E-mail: yhliu@ccnu.edu.cn
[2)]College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074 , China

ABSTRACT

The goal of this paper is to research the topology discovery of a computer network algorithms. Some key techniques of backbone topology discovery are discussed in detail, and some solutions are proposed, such as how to ascertain the link relations of network equipments, how to mark a router, how to avoid a router being accessed repeatedly, and how to distinguish a router from gateway. This paper also analyses the principle of subnet topology discovery and presents a method for discovering the live hosts in a subnet. On these basis, a backbone topology discovery algorithm and a subnet topology discovery algorithm are designed by ourselves, and the complexity of these algorithms are analysed. At last all the algorithms are implemented successfully in object oriented language-JAVA2.

Keywords: network topology, backbone, subnet, discovery algorithm, SNMP, ARP.

1 . INTRODUCTION

The network topology discovery is one of the basic components of network management functions[1].Its goal is to find out the dynamic link relations and logic topology relations among all kinds of equipments in the network by searching in real time, and show them out in graphics mode directly and explicitly, so as to make network management operator know correctly the state in which network runs, and deal with the network management functions such as configuration, fault, performance, security and accounting better. Consequently, it is particularly important to develop and design high-efficient discovery algorithms, and combining many kinds of topology discovery algorithms could guarantee the automatic topology discovery fast, the overhead on the network low, the topology map complete and without mistakes[2].

At present some network management products can search the network automatically, and make relatively good results, but they are not as good as our expectation[3]. This paper gives out two algorithms which are designed by ourselves. One is based on SNMP of backbone topology discovery and the other is based on ARP of subnet topology discovery. They rectify some weakness of the present algorithms and achieved a relatively satisfying effect.

2 . BACKBONE TOPOLOGY DISCOVERY

The most promising tool is Simple Network Management Protocol(SNMP).Every device has a Management Information Base(MIB),which stores all the information about the processes that is running. Backbone topology discovery is to search for the relations of links among the key equipments including routers, ports and subnets (figure 1), Namely the router - router, router - subnet relations. At first it gets some systematic data used for network topology discovery，such as seed node, community name, etc., to set up target nodes. If it is necessary not to collect new information, it will get topology data directly form data base or adjacent tables to set up target joints. When initializing the systematic data, it is easy to obtain original IP address of seed node and router hops for searching.

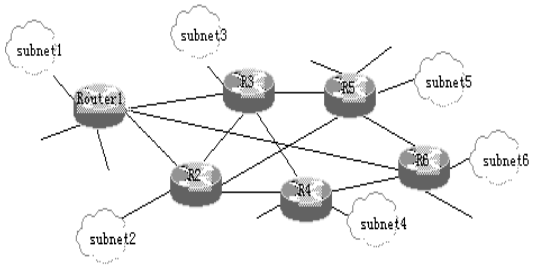


Figure 1 Network containing routers and subnets

2.1 The key technologies of Backbone topology discovery

The backbone topology discovery mainly solves the key techniques as follows:

1) How to distinguish a router from a gateway

The key equipments such as routers, switches, intelligent HUBs. Generally have SNMP to act as their agent. The type of the network equipment can be retrieved by visiting its MIB to check ipForwarding and sysServices variables. The variable ipForwarding is used to judge whether the equipment can forward ip packets, furthermore whether it is a gateway or not.

2) How to distinguish subnets from routers (or gateways) by the next hops in routers

When using interface table-ifTable of network equipments, at first we visit the address table of the device and get the network address and
interface index of each device. Then according to the interface index, we inquire the interface table of the device and get the type of interfaces by which this equipment connecting with the other network equipments. Through the interface type, network equipments can be judged further. We know that ifDesr and ifType is used to name the interface and give its type respectively. For example, if subnet is Bus, ifType is 6, and ifType is 9 for Token-Ring.

3) How to confirm the link relation between router and router, router and subnet

The ipRouteType variable denotes the type of a router record, in the IP router tabel (ipRouteTable) of a network equipment. With the value of 3,it represents the corresponding interface of this router links to a subnet directly and value 4 represents this interface links to another router or gateway.

4) How to mark a equipment uniquely

As to mark a network equipment(including router, switch, intelligent HUB and host), it has been put forward that these equipments can be signed with IP because a router has many IP addresses, the smallest IP can be used to sign the router. This kind of method is feasible, but it has some weakness too: In order to get the smallest IP, all the Ip addresses of a router should be compared with each other and CPU time will be consumed in this procedure. For this reason, each router can be specified a global index number. At the same time, in order to distinguish the different kinds of equipments, such as router, switch, host, etc., we can specify different equipment_type_tags to different equipments.

5) How to prevent the same router from being visited epeatedly

In order to prevent the same router from being visited repeatedly, we can set up a accessed_IP_queue. When a router is accessed through one of its IP, put all the routers' IPs into the accessed_IP_queue. Before accessing a router through a IP address, we search the accessed_IP_queue firstly to see whether this is IP in the accessed_IP_queue. If it exists, jump over this IP, or try to access the corresponding router through this IP.

2.2 A algorithm for Backbone topology discovery

After solving above key problems, we present a algorithm based on SNMP. It will be described as algorithm 1.

algorithm 1:

Initialize link_relation_queue which stores the relationships between router and router,router and switch,router and subnet；
Initialize router_to_access_queue (storing the router entities to be accessed),IP_accessed_queue (storing the IP address that need not to be accessed), and subnet_accessed_queue (storing the subnets that need not to be accessed) by using seed IP；

```
while (router_to_access_queue is not empty)
{
    get one router entity from router_to_access_queue as
    CurrentRouter；
    access CurrentRouter  routetable，for each item of
    the route table：
   {
     if (ipRouteType==indirect)
     {
         if(ipRouteNextHop is in IP_accessed_ queue)
         put the relationship between CurrentRouter and
         the ipRouteNextHop into link_relation_queue;
       else
       {
           initialize the router of ipRouteNextHop,put it
           into router_to_access_queue, put the
           relationship between CurrentRouter and the
           router into link_relation_queue;
       }
     }
    else if (ipRouteType==direct)
    {
     if(ipRouteDest is in subnet_accessed_queue)
          put the relationship between CurrentRouter and
          the subnet of netAddr into link_relation_queue;
    else
     initialize the router according to
     ipRouteNextHop,put it into
     router_to_accessed_queue, put the relationship
     between CurrentRouter and the router into
     link_relation_queue;
    }
  }
```

2.3 The analysis of complexity of algorithm 1

We suppose the number of each router's ipRouteEntry is D. The number of the routers connecting with one router is Rc(quantity class is 10). The number of subnets connecting with the same router is Sc(quantity class is 10). The total number of routers that can be visited is R, subnetworks that can be visited is S and links of the network is L. Usually D>>Sc, R>>Rc. Correspondingly the total complexity is $O(R*logR)+R*(O(D)*O(Rc)+O(Sc)*O(Sc)) \approx O(R^2)$. As for the backbone topology discovery algorithm based on SNMP, no matter taking which algorithm, all the route table of each router must be accessed, the complexity of the algorithm is at least $O(R*D)=O(R^2)$,So there is no other algorithms whose complexity is lower than this algorithm. When the total amount (R) of routers which can be managed is very large, the complexity become O(R*LogR), under this condition a lot of routers must be accessed, the time expended on running will be too much. So it is necessary to limit the discovery hops or the range of IP to be accessed.

3. SUBNET TOPOLOGY DISCOVERY

Generally there are two kinds of methods to discover the live equipments of a subnet: one is based on ARP, and the other is based on ICMP. For those live equipments, deliver SNMP message to port 161 of each host to get the value of the MIB variable sysUpTime. If the value is not empty, the equipment has run the SNMP agent, and its MIB should be accessed to get more detailed information.

3.1 The principle based on ARP of subnet topology discovery

In order to reduce the overhead, the computer which has ARP table maintains a high speed buffer memory to store the mapping from IP address to physical address. We can get the Ip address one by one through accessing the MIB variable ipNetToMediaNetAddress, then make "AND" operation with each IP address and corresponding mask to get a subnet address. If this subnet address is the same as the specified subnet address, the host with this IP address belongs to the subnet.

This system adopts the method based on ARP to discover a subnet mainly for the following reasons:(1) This method can utilize the address conversion table of the router to get the information of live hosts in the subnet directly, and reduce the time greatly compared with the traditional PING method;(2) It will not increase more network overheads;(3) Implement easily;(4) This system is realized with pure "JAVA" in order that it can be run on any platform. See the algorithm in the next section.

3.2 A algorithm for subnet topology discovery

Agorithm 2:

step1 Initialize the equipment_accessed_ queue which used to store all live devices in a subnet, such as routers, switches, hosts, etc.;

step2 Initialize IP_to_accessed_ queue;

step3 Get the subnet address (netAddress) and subnet mask (mask);

step4 Get a router linked to the subnet directly, note it as relateRou- er,and put this router into equipment_accessed_ queue;

step5 Access the relateRouter's MIB, get the value of

variable ifType;

step6 If there is variable ipNetToMediaPhysAddress that has not been accessed, get its value, a equipment's IP; else go to step10;

step7 Do the 'AND' operation of mask and IP gotten from step6, get a network address;

step8 Compare the network address gotten from step7 with netAddress, if equal, put the IP gotten from step6 into IP_to_accessed_ queue;

step9 Go to step6;

step10 If IP_to_accessed_ queue is not empty, get the first one;

step11 Send SNMP message to the equipment with the IP gotten from step10, get the value of MIB variable sysUpTime, if it is not empty, go to step12, else go to step13;

step12 Get the value of MIB variable sysServices to judge the equipment a router or a switch and get more other information, go to step14;

step13 Generate a host entry, and put it into equipment_accessed_ queue;

step14 If IP_to_accessed_ queue is not empty, get the next one and
go to step11, else go to step15;

step15 End.

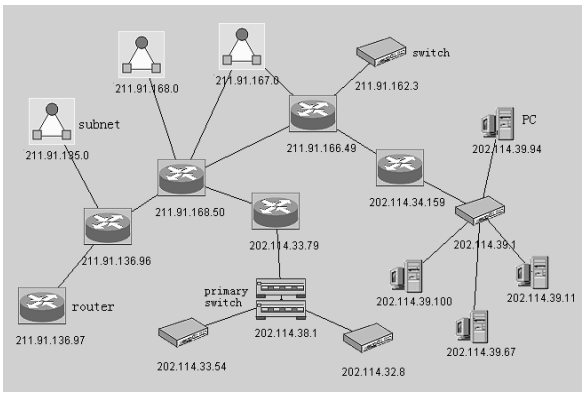


Figure 2 A portion of the campus network topology discovered by our algorithms

If adopting the above mentioned method to discover a subnet, the time for searching is in ratio to the number of live equipments. A standard network of class C can be discovered in four seconds. When the live equipments of a subnet are few, it will waste much time to wait the responding ICMP packets with a PING algorithm.

4. CONCLUSIONS

We have implemented the backbone topology discovery algorithm and the subnet topology discovery algorithm with JAVA 2. The figure 2 shows a network topology discovery instance in our CCNU campus network.. Also there are some kinds of backbone topology discovery algorithms based on OSPF, RIP or IP-TTL at present, but our algorithm is easier to implement and impose the least possible overhead on the network. In order to get rid of the redundant information and make our concern more centralized, we can set a seed IP address and a router hop to control the backbone topology discovery scale, or constraint the IP address range for discovering.

5. REFERENCES

[1] LEI Zhengjia.Computer Network Management and System Develepment. BEIJING. Publishing House Electronics Industry. 2002. 1

[2] Douglas E.Comer. Linyao etc.,Interpret. Internetworking withTCP/IP Vol Ⅰ: Principles, Protocols, and Architectures Fourth Edition. BEIJING. Publishing House Electronics Industry. 2002. 5

[3] Bai yingcai, etc.,The Design and Apply of Computer ManagementSystem. BEIJING. Publishing House Tsinghua University,1998.9

[4] R.Siamwalla, R.Sharma, and S.Keshav. Discovering Internet Topology. IEEE INFOCOM 1999

[5] W.Richard Steven. TCP/IP Illustrated Volume 1: The ProtocolsAddison-Wesley,1994

[6] Li Jia, Shi Binxin, etc.Automatic Topology Discovery Algorithm for Network Management System. Journal of Huazhong Univ. of Sci.&Tech, Vol 26 No.1,Jan. 1998

[7] Xu Dahai,Liu Xin, etc. Algorithm of Network Topology Search. Computer Applications, Vol.19,No.2,Feb. 1999

* The research is supported by the grant of Natural Science Foundation in Hubei Province (2001ABB013) and Key Science & Technology Foundation in Hubei Province (2001AA104A05).

A Robust Level Set Algorithm for Image Segmentation and its Parallel Implementation

Joris R. Rommelse*
Department of Applied Mathematics Delft University of Technology
Delft, 2628 CD, The Netherlands
E-mail: J.R.Rommelse@its.tudelft.nl
Hai-Xiang Lin
Department of Applied Mathematics Delft University of Technology
Delft, 2628 CD, The Netherlands
E-mail: H.X.Lin@its.tudelft.nl
Tony F. Chan
Department of Mathematics University of California, Los Angeles
Los Angeles, CA 90095-1555, USA
E-mail: Chan@math.ucla.edu

ABSTRACT

In this paper we discuss a classic clustering algorithm that can be used to segment images and a recently developed active contour image segmentation model. We propose integrating aspects of the classic algorithm to improve the active contour model. For the resulting CVK segmentation algorithm we examine two methods to decrease the size of the image domain.
The CVK method has been implemented to run on parallel and distributed computers. By changing the order of updating the pixels, it was possible to replace synchronous communication with asynchronous communication and subsequently the parallel efficiency is improved.

Keywords: Image Segmentation, Clustering Algorithm, Active Contour Model, Level Set Functions, Synchronous and Asynchronous Communication.

1. INTRODUCTION

Looking at an image, it is usually very easy for a human to see what it represents. For a computer this is not so easy. Before computers "know" what is represented, objects must be measured, but before that, the objects must first be detected. Understanding images is very important in problems like stereo and motion estimation, part recognition or image indexing. The first step in image understanding is image segmentation. Segmentation is the problem of dividing an image into objects or distinguishing objects from a background.

This paper discusses the classic K-means algorithm, the recently developed Chan - Vese (CV) active contour model and a combination of both (CVK). K-means can only handle a small subset of images and needs image enhancement for noisy or blurred images. CV and CVK are designed to handle a much larger class of images without enhancement. Unfortunately these algorithms are often slower. Without any optimizations, a typical 600×480 image will take several hours. Therefore methods that reduce the size of the image domain and parallel and distributed computers are used to speedup calculation.

Section 1 introduces image segmentation and the K-means, CV and CVK algorithms. In section 2 a narrow band version and a multiresolution version of the CVK algorithm are examined to decrease the size of the image domain. Section 3 discusses the parallel implementation of these versions and shows experiments with synchronous and asynchronous inter-processor communication. Section 4 gives some general conclusions.

Image Segmentation

For every point in an image domain, $x \in \Omega$, the intensity of a grey-valued image can be specified by a number in [0,1] (0 for black, 1 for white) and the intensity of a RGB image (red, green, blue) can be specified by three numbers in [0,1]. In general, the intensity of images considered in this paper can be specified by m numbers in [0,1]. Therefore images are mappings from Ω to $[0,1]^m$ and can be written as

$$u : \Omega \to [0,1]^m \quad , \quad u(x) = (u_0(x), \cdots, u_{m-1}(x)) \quad , \quad x \in \Omega \qquad (1)$$

The image domain can be discrete (a grid of points or pixels) or continuous, has a rectangular shape and can have any number of dimensions larger than one.
For example,

$$\Omega = [0, w_0] \times [0, w_1] \qquad (2)$$

for a 2D continuous image, or

$$\Omega = \{0, \cdots, \omega_0 - 1\} \times \cdots \times \{0, \cdots, \omega_{d-1} - 1\} \qquad (3)$$

for a d-dimensional digital image. Here w_0 and w_1 are the width and height of the image and ω_i is the number of grid points in the (i+1)-th dimension.

The goal of image segmentation is to segment the image domain Ω into several subdomains Ω_i, based on some appropriate criteria that involve intensity or location of colors, such that the domain is formed by the union of the subdomains and the subdomains do not overlap,

$$\Omega = \bigcup_{i=0}^{k-1} \Omega_i \quad , \quad \Omega_i \cap \Omega_j = \varnothing \quad (i \neq j). \qquad (4)$$

After segmentation is completed, the subdomains are considered k separate objects, or k-1 separate objects and a background. The objects can then be measured and classified or recognized. This step towards image understanding is not covered by this paper.

Some Commonly Used Segmentation Algorithms

(1) K-means clustering algorithm [6]

* Supported in part by Universiteitsfonds Delft

The K-means clustering algorithm was designed to cluster a number (N) of objects into K clusters or classes, based on location of the objects. The objective is to assign the objects to classes so that they lie closer to the average location of their class than to the average location of other classes.

The algorithm can be used to cluster pixels of a digital image with related color if the pixels are considered objects. Location is measured in terms of color rather than the actual position of the pixels in the image; pixels are located in color space, not in physical space. The average location in color space of pixels in a class can be calculated by averaging the colors of all pixels in the class. The distance between a pixel p_i and the average of a class a_j, can be expressed by

$$d(p_i,a_j)=\tfrac{1}{m}\sum_{k=0}^{m-1}\lambda_{j,k}(p_{i,k}-a_{j,k})^2\,, \tag{5}$$

(m=1 for a grey valued image and m=3 for a RGB image).
The parameters $\lambda_{j,k}$ can be used to give priority to classes or colors.

(2) Chan - Vese image segmentation [4,10,11]
A very different approach is segmentation by curve evolution, snakes or active contour models. Here a parameterized hypersurface C (or contour in 2D) moves through the image domain with respect to constraints from the image and stops on the boundaries of objects. Mumford & Shaw base the constraints on the minimization of an energy functional

$$F^{MS}(\bar{u},C)=\mu\cdot surface(C)+\lambda\int_{\Omega}|u-\bar{u}|^2\,\mathrm{dx}+\int_{\Omega\setminus C}|\nabla\bar{u}|^2\,\mathrm{dx} \tag{6}$$

where **u** is the original image, $\bar{u}$ is the segmented image, C is a hypersurface around detected objects, *surface*(C) is the length of the contour (in 2D) or the area of the surface or hypersurface (in 3D or higher dimensions), $\mu\ge 0$ and $\lambda\ge 0$ are parameters that can be set by the user.

Chan & Vese represent the hypersurface C by a set of n simple closed hypersurfaces $\{C_0,\cdots,C_{n-1}\}$,

$$C=\bigcup_{i=0}^{n-1}C_i\,. \tag{7}$$

Every hypersurface C_i segments Ω/C in 2 subdomains and the set of n hypersurfaces therefore segments Ω/C in $2n$ subdomains $\{\Omega_0,\cdots,\Omega_{2^n-1}\}$.

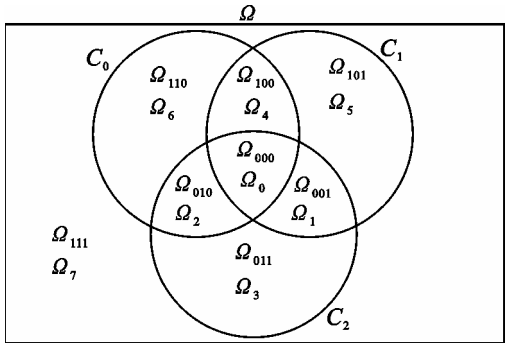


Figure 1 Segmentation of the image domain Ω in eight subdomains $\Omega_0,\ldots,\Omega_7$ by three hypersurfaces C_0,C_1,C_2.

The binary representation of the index i of subdomain Ω_i can be found by the relative position to the hypersurfaces. Example: Ω_{001} lies outside (1) C_0, inside (0) C_1 and inside (0) C_2.

Furthermore, they restrict $\bar{u}$ to piecewise constant functions; $\bar{u}$ has a constant value $\boldsymbol{c}_i$ on every subdomain Ω_i

$$\bar{u}(x)=c_i \text{ if } x\in\Omega_i$$
$$\bar{u}(x)=\sum_{i=0}^{2^n-1}c_i\chi_i(x) \tag{8}$$

with

$$\chi_i(x)=\begin{cases}1 & \text{if } x\in\Omega_i\\ 0 & \text{if } x\notin\Omega_i\end{cases} \tag{9}$$

the characteristic function of subdomain Ω_i.

Now the Mumford-Shaw energy functional Eq. (6) becomes

$$F_n^{MS}(C_0,\cdots,C_{n-1},c_0,\cdots,c_{2^n-1})=\\ =\mu\cdot surface\left(\bigcup_{i=0}^{n-1}C_i\right)+\sum_{i=0}^{2^n-1}\lambda_i\int_{\Omega_i}|u-c_i|^2\,\mathrm{dx} \tag{10}$$

For vector-valued images (m>1), the Chan-Vese energy functional is defined by

$$F_{n,m}^{CV}(C_0,\cdots,C_{n-1},c_0,\cdots,c_{2^n-1})=\\ =\mu\cdot\sum_{i=0}^{n-1}surface(C_i)+\tfrac{1}{m}\sum_{i=0}^{2^n-1}\sum_{j=0}^{m-1}\lambda_i\int_{\Omega_i}(u_j-c_{i,j})^2\chi_i\mathrm{dx} \tag{11}$$

The steps of the algorithm are to minimize the energy by moving the hypersurfaces while keeping $\{c_0,\cdots,c_{2^n-1}\}$ fixed, and then recalculate $\{c_0,\cdots,c_{2^n-1}\}$ while keeping the hypersurfaces fixed. The latter can be done by setting the derivatives with respect to $c_{i,j}$ to zero:

$$\frac{\partial F_{n,m}^{CV}}{\partial c_{i,j}}=0\quad\Rightarrow\quad c_{i,j}=\frac{\int_{\Omega}u_j\chi_i\mathrm{dx}}{\int_{\Omega}\chi_i\mathrm{dx}}=\frac{\int_{\Omega_i}u_j\mathrm{dx}}{\int_{\Omega_i}\mathrm{dx}}=\frac{\int_{\Omega_i}u_j\mathrm{dx}}{|\Omega_i|} \tag{12}$$

In other words, $\boldsymbol{c}_i$ is the average color of the image on subdomain Ω_i.
For the former, an appropriate representation for the hypersurfaces must be chosen. Chan & Vese choose to use level set functions. On a d-dimensional domain Ω, a hypersurface C is a (d-1)-dimensional object, but it can be represented by a function on Ω, $\phi:\Omega\to R$.

$$C=\{x\in\Omega:\phi(x)=0\} \tag{13}$$

C is the zero level set of ϕ and $n(x_0)=\frac{\nabla\phi}{|\nabla\phi|}$ is a vector normal to $\{x\in\Omega:\phi(x)=\phi(x_0)\}$ at position $\boldsymbol{x}_0$ pointing in the direction of level sets with $\phi(x)>\phi(x_0)$.

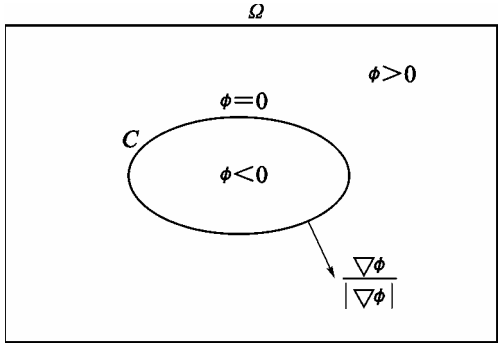


Figure 2 Representation of a hypersurface C and the unit normal by a level set function ϕ on Ω.

Moreover, the position of a moving hypersurface can be calculated by solving an evolution equation for ϕ:

$$\phi_t + v|\nabla\phi| = 0 \quad , \quad \text{given } \phi(\mathbf{x},t) \tag{14}$$

Not just the zero level set, but all the level sets move in normal direction with speed $v(\mathbf{x},t)$.

To segment the image, a suitable speed function $v(\mathbf{x},t)$ can be derived for every hypersurface C_i:

$$\frac{\partial\phi_i}{\partial t} = |\nabla\phi_i| \left(\mu\nabla\cdot\left(\frac{\nabla\phi_i}{|\nabla\phi_i|}\right) - \text{sign}(\phi_i)(p_I - p_{J_i}) \right) \tag{15}$$

Here $I(\mathbf{x})$ is the index of the subdomain where $\boldsymbol{x}$ lies, $J_i(\boldsymbol{x})$ is the index of the subdomain that lies opposite subdomain $I(\boldsymbol{x})$ relative to hypersurface C_i and $p_i : \Omega \to R$ are penalty functions to express why points in Ω should not be in Ω_i,

$$p_i(x) = \tfrac{1}{m}\sum_{j=0}^{m-1} \lambda_{i,j}(u_j - c_{i,j})^2 \,. \tag{16}$$

Notice that these penalties are also used in the K-means algorithm.

In digital image processing, the PDE Eq. (15) is discretized using central differences for the spatial derivatives and Euler forward for the time derivatives, to fit the given grid. This means that for stability reasons, the time step must be chosen depending on the given spatial step.

(3)C VK algorithm [8]

In the CV algorithm, the color of a pixel is compared to the mean color of its subdomain and the mean colors of the subdomains that lie opposite the hypersurfaces. Therefore a pixel can stay in its subdomain or move to one of n other subdomains (if there are n hypersurfaces). However, the K-means algorithm allows pixels to move to any of the other $2n$-1 subdomain. Apparently in the CV algorithm, pixels might be denied the opportunity to move to the right subdomain.

The CVK algorithm segments an image by evolving hypersurfaces according to the PDE

$$\frac{\partial\phi_i(x)}{\partial t} = |\nabla\phi_i(x)| \left(\mu\nabla\cdot\left(\frac{\nabla\phi_i(x)}{|\nabla\phi_i(x)|}\right) - (1-\mu)\text{sign}(\phi_i(y)) \right) \tag{17}$$

where $\mu \in [0,1]$ and

$$\text{y} = \underset{z\in\Omega}{\text{argmin}} \left\{ \tfrac{1}{m}\sum_{j=0}^{m-1} \lambda_{i,j}(u_j(z) - c_{i,j}(z))^2 \right\} \tag{18}$$

is a pixel in the subdomain where the penalty function Eq. (16) is minimized, so where a pixel $\boldsymbol{x}$ should be moved to according to the K-means algorithm. Although there are many such pixels $\boldsymbol{y}$, the level set function ϕ_i has the same sign on all of them. The sign function in Eq. (17) makes sure that the level set function ϕ_i on pixel $\boldsymbol{x}$ gets closer to zero or even changes sign when $\boldsymbol{x}$ is located on the wrong side of hypersurface C_i. In case $\boldsymbol{x}$ is located on the right side of C_i, ϕ_i is updated such that $|\partial\phi_i/\partial t| > 0$. This is done so the color criterion can oppose the curvature criterion that will be discussed in the sequel, to prevent the hypersurface from showing wiggling behavior when these criteria contradict and alternate dominance in subsequent iterations.

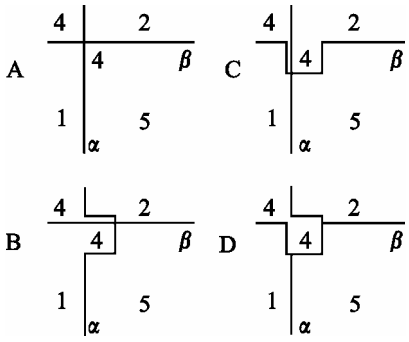


Figure 3 Near intersections of hypersurfaces, the CV algorithm might fail where the K-means algorithm does not.

The figure shows a hypothetical situation where four classes/subdomains are separated by two hypersurfaces α and β. The averages in the classes are 4,1,2 and 5. A pixel/object with value 4 is currently assigned to the class with average 5 (A) and should be assigned to the class with average 4 (D), which means that both hypersurface α (B) and hypersurface β (C) will have to move. For the obvious choice of parameters $\lambda_{i,j}$, the CV algorithm does not move the hypersurfaces, because $(4-5)^2<(4-1)^2$ and $(4-5)^2<(4-2)^2$. The K-means algorithm does move the hypersurfaces, because $(4-5)^2>(4-4)^2$.

Qualitative Evaluation Of The Algorithms

For undamaged, unblurred, synthetic images, all three algorithms (K-means, CV and CVK) work well. For natural images or noisy images, the K-means algorithm cannot be used to completely segment the images [6,8], although it can still be useful to create an initial guess for other algorithms. CV and CVK are designed to handle these images as well [4,8,10].

Because

$$n = \frac{\nabla\phi_i}{|\nabla\phi_i|} \tag{19}$$

is a unit normal to hypersurface C_i,

$$\kappa = \nabla\cdot\left(\frac{\nabla\phi_i}{|\nabla\phi_i|}\right) \tag{20}$$

can be used to calculate the curvature κ of the hypersurface. The hypersurfaces are moved by two effects; a fitting term makes sure that pixels in the same object have similar color and a curvature term makes the contours move in the direction that minimizes the curvature, $\frac{\partial}{\partial t}|\kappa| \le 0$. Small objects have large curvature, while large objects have smaller curvature. Noise is actually made of many very small objects and have large curvature. The parameter μ can be set by the user to specify whether large or small objects should be detected and can be used to make the algorithms (CV and CVK) robust to noise. The curvature term deals with noise and keeps detected objects from being scattered.

The improvement in quality of the CVK over CV can not be measured in terms of correct output of the algorithms, but in terms of user friendliness; both algorithms produce correct results if the right set of parameters μ and $\lambda_{i,j}$ are put in. However, it can be tricky to tune the parameters μ and $\lambda_{i,j}$ for the CV algorithm, whereas CVK works fine by just choosing μ and setting all parameters $\lambda_{i,j}$ equal to one [8,10]. So the CVK algorithm is more robust in usage.

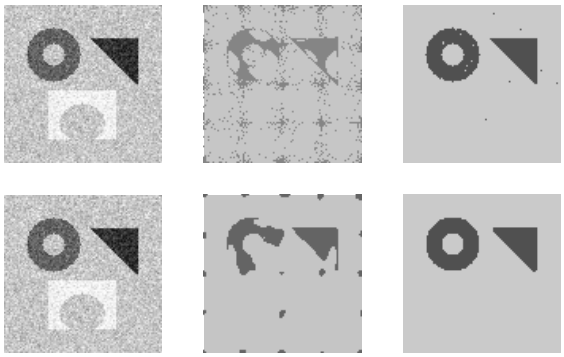


Figure 4 From left to right: noisy input image, segmentations after some iterations and final segmentation.
Top: only K-means criterium was used. Bottom: both K-means and curvature criteria were used. CVK: $\lambda_0 = \lambda_1 = 1$, CV: $\lambda_0 < \lambda_1$ [6].

2. CVK ALGORITHM

Smaller Domain Versions
(1) Full domain
Choosing level set functions to represent hypersurfaces introduces the flexibility that is so much needed, because hypersurfaces may split or merge while moving. On the other hand, they come with extra calculation time, because d-1 dimensional objects are represented by functions on a d dimensional domain.

(2) Narrow band
The first step in reducing the time complexity is to acknowledge that much work is done in vain. In the digital model, a hypersurface moves over a pixel if the level set function on that pixel changes sign in the iteration. Assuming that far away from the hypersurface this changing sign does not happen, it is a waste of time to update the level set function there using the evolution equation. However, this assumption might not always be justified. Instead of updating the level set functions on every grid point, a speedup will be achieved if only values on grid points near the hypersurfaces are updated. By specifying a maximum distance δ to the contours and only updating the level set functions on grid points within this distance, a band-shaped domain is created.
Applying the narrow band method to the segmentation algorithm may cause the algorithm to fail. The narrow band method will produce correct results if the speed in normal direction depends only on local properties like the curvature. The fitting term may cause new hypersurfaces to appear out of nothing. This means that new hypersurfaces that should appear more than δ away from existing hypersurfaces do not get a chance in the narrow band method. Or objects that are not yet detected might not be detected at all if they are located too far from objects that are already detected.
The narrow band method can still be used with the segmentation algorithm if the initial hypersurfaces are chosen well. The algorithm can be expected to succeed if the union of the narrow bands corresponding to the initial contours cover most of the image domain Ω. In that case no speedup can be expected in the first few iterations after initialization.
Here the location of the narrow band is stored, along with the location of the zero level set, by the level set function; the edges of the band are the δ and -δ level sets. Reinitialization is needed to keep the distance between the level sets constant. In [1] a data structure is built that can store the location of the band during more than one iterations.
(3) Multiresolution
Decreasing the resolution of an image decreases the size of the image domain and thereby reduces the time complexity of the segmentation algorithm. Changing the resolution of a digital image means that the same image is spread over a different number of pixels. A multiresolution method can take advantage of this. The multiresolution method should not be confused with the standard multigrid method, in which an iterative solution and the corresponding problem are coarsened to another grid, where the problem is solved and interpolated back to the fine grid. The grids are used recursively and iteratively. The multiresolution method for the image segmentation problem uses lower resolution versions of the original image to find initial solutions for higher resolution problems instead. So where the multigrid method starts at the highest level, returns to the highest level and uses all coarse grids regularly, the multiresolution method starts at the lowest level, ends at the highest level and uses all coarse grids only once.
The only required addition is a mechanism that can resize a d dimensional grid. If d=1, a value on a new grid point can be calculated by linear interpolating the values on the neighbor grid points in the old grid (for the level set functions) or by copying the value on the nearest neighbor grid point of the old grid to the new grid point. If d>1, this mechanism is used for every dimension.

Quantitative Evaluation Of The CVK Versions
In the narrow band method, time is saved because calculations are only performed on a small domain. On the other hand, extra administration is needed to calculate and store the location of the narrow band. In the current implementation, the narrow band does not result in speedup but some speeddown, whereas previous versions of the narrow band did result in speedup. This is not a flaw in the current implementation of the narrow band method. In previous implementation of the full domain method, the level set methods had to be reinitialized after every iteration. This could be eliminated in the current implementation of the full domain method, but not in the implementation of the narrow band method.
The multiresolution method does not only reduce the number of grid points, but also reduces the number of operations that have to be performed on every grid point. For stability reasons, $\tau = O(h^2)$, with τ the time step and h the spatial step. On a coarser grid, larger time steps can be made, so lesser iterations are needed.

3. PARALLELIZATION

Parallel CVK Algorithm
In the sense of tasks that have to be performed, the segmentation algorithm is clearly sequential by nature. Therefore a data parallel model of computation is chosen. The first dimension of the domain is partitioned while the other dimensions are not partitioned. So if $\Omega = \{0,\cdots,\omega_0 - 1\} \times \cdots \times \{0,\cdots,\omega_{d-1} - 1\}$, then $\{0,\cdots,\omega_0 - 1\}$ is partitioned into S subsets $\{0,\cdots,\omega_0 - 1\} = \{\varpi_0,\cdots,\varpi_1 - 1\} \bigcup \cdots \bigcup \{\varpi_{S-1},\cdots,\varpi_S - 1\}$, where

$$\varpi_i = i \left\lfloor \frac{\omega_0}{S} \right\rfloor + \min\{i, \omega_0 \bmod S\} \tag{21}$$

and

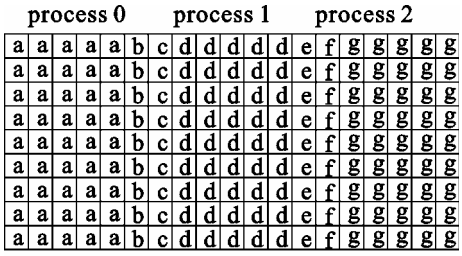
$$\overline{\Omega}_i = \{\varpi_i, \cdots, \varpi_{i+1} - 1\} \times \{0, \cdots, \omega_1 - 1\} \times \cdots \times \{0, \cdots, \omega_{d-1} - 1\} \quad (22)$$

If $d = 2$ this is called striped partitioning; the domain is a matrix and the columns are assigned to sub-matrices. In 3D it could be called sliced partitioning. To let terminology stay valid for $d > 2$ let

$$\{i\} \times \{0, \cdots, \omega_1 - 1\} \times \cdots \times \{0, \cdots, \omega_{d-1} - 1\} \quad (23)$$

be the (i+1)-th column of Ω. The first $\omega_0 \bmod S$ subdomains get $\lfloor \omega_0 / S \rfloor + 1$ columns and the other subdomains get $\lfloor \omega_0 / S \rfloor$ columns.

The grid points of the subdomains are assigned to S processes, a processor can run more than one process. Assigning a grid point to a process means that all data associated with that grid point is stored in the memory of the processor on which the process will run and that operations on the data will be performed by that processor. For every column that does not belong to the same process as its neighbor column an extra column is assigned; operations on this data are done by the neighbor process and data is updated during a synchronizing step between processes. In the synchronization step data is sent between processes, although sending is not the correct word if both processes are located on the same processor. This mechanism makes sure that subdomains overlap by one column, which is exactly enough for the scheme used in the discrete evolution equation.



process	calculations	data
0	a b	a b c
1	c d e	b c d e f
2	f g	e f g

Figure 5 2D domain distributed to three processes.
Data associated with a subdomain is assigned to the concerning process and calculations needed to update the data are done by the process. Columns of data from neighbor processes are needed, so synchronization must take place.

Except after initialization, the hypersurfaces will be close to the objects to be detected. Since they are not distributed uniformly over the image, neither will be the narrow bands. Some parts of the domain may require more calculation than other parts. To distribute the work equally among the p processors, the S subdomains are mapped cyclically to the processors. Given p, S must be chosen so that $S \bmod p = 0$.

Let b=S/p, then the domain is divided into b blocks and every block is distributed over p processors. So if p is given, b can be chosen small to reduce communication between processors or b can be chosen large to ensure a good load balance. In case no narrow band is used, the load is well balanced automatically, so b can be chosen equal to one.

Synchronous And Asynchronous Communication

Before every iteration the overlapping data must be synchronized. This requires S-1 exchanges of columns between neighbor processes. Also two all-to-all broadcasts must take place to evaluate the stop criterion and to recalculate the average intensities $c_{i,j}$.

Subdomains $\overline{\Omega}_i$ are assigned to processor $i \bmod p$. If neighbor processes are not located on the same processor, which is the case for every process if $p > 1$, synchronization must take place. This takes either two or three steps, depending on whether p is odd or even. In every step a process communicates with either its left or its right neighbor process. Here the communication time is minimized by bundling b overlapping areas into one message. Messages do not actually have to be copied to a combined message; instead computer memory can be organized so blocks of data that have to be sent in the same communication step are always located at consecutive addresses.

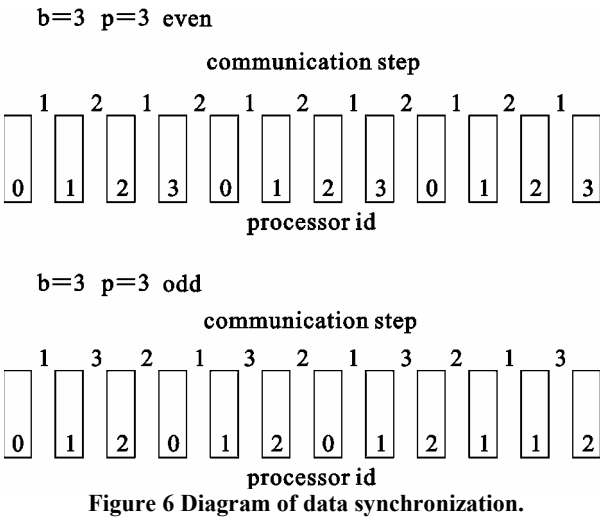


Figure 6 Diagram of data synchronization.
Top: 12 processes are mapped to 4 processors.
First step: communication between 0 and 1 and between 2 and 3.
Second step: communication between 0 and 3 and between 1 and 2.
Bottom: 12 processes are mapped to 3 processors.
First step: communication between 0 and 1.
Second step: communication between 0 and 2.
Third step: communication between 1 and 2.

(1) Asynchronous communication

On parallel computer systems where communication is slow, run time could be optimized if calculations and communication could be done in parallel. The advantage of communicating asynchronously is that processors do not have to wait until the other processor involved in the communication is ready. The disadvantage is that after the communication is initiated, the data involved is accessible but not yet usable. This can be dealt with by choosing the order in which grid points are updated carefully: first non-overlapping grid points must be updated, then the overlapping grid points.

Iteration with synchronous communication:

- initiate synchronous communication of overlapping data
- wait until communication is finished
- update overlapping grid points
- update non-overlapping grid points

Iteration with asynchronous communication:

- initiate asynchronous communication of overlapping data
- while communication is in progress, update non-overlapping grid points in parallel

- wait until communication is finished
- update overlapping grid points

In MPI implementations, synchronous and asynchronous receive operations are implemented by the functions MPI_Recv and MPI_Irecv respectively. Synchronous and asynchronous send operations are implemented by the functions MPI_Send and MPI_Isend. MPI_Isend is in fact asynchronous, but MPI_Send can be synchronous or asynchronous, depending on the size of the systems message buffer; if the buffer is large enough, asynchronous sending is used.
Because some versions of MPI always use asynchronous sending, sometimes few improvement can be observed by using MPI_Isend and MPI_Irecv instead of MPI_Send and MPI_recv.

Results
The CVK algorithm was implemented (full domain, narrow band and multiresolution) in C++ using MPI functions for communication.
For undamaged, unblurred, synthetic images, K-means can be expected to be very fast. The algorithm can finish in just a few iterations, because in these kind of images there is only a small amount of different colors and pixels are assigned to classes based on color and not on location: if it is decided that a pixel with some color should be in some class, than all pixels with the same color are assigned to that class in the same iteration. CV and CVK on the other hand are slower, because only pixels near hypersurfaces are reassigned. For natural images or noisy images, the speed of the K-means algorithm is irrelevant, because the algorithm is not applicable.
Calculation time for CV is discussed in [10]. For the CVK algorithm, calculation time was measured on a Cray T3E parallel computer [13] in Delft and the DAS-2 clusters [14]. The efficiency of the algorithm run on DAS-2 on several grey-valued and color images, ranging in size from 100×100 to 600×480, using one or two level set functions, using different initializations and using different numbers of coarse grids, is shown is figure 7. Improvement in efficiency can be observed for most test cases on DAS-2, if asynchronous communication is used. Figure 8 shows an example of typical improvement by the asynchronous version. For large images, much less improvement can be seen, but for these images communication overhead plays a less important role relative to the increased number of calculations. Because both MPI_Send and MPI_Isend are implemented asynchronously with default setting of MPI_BUFFER_MAX on the Cray T3E, the use of MPI_Send and MPI_Isend do not influence the efficiency much.

4. CONCLUDING REMARKS

In this paper we discussed the parallel implementation of a new image segmentation method (CVK), a method that was created by integrating a classic clustering algorithm (K-means) into a recently developed active contour model (Chan - Vese). The narrow band method and the multiresolution methods were attempts to decrease the size of the image domain and thereby the calculation time. The multiresolution method proved very useful for regular images and indispensable for large images. Because of reinitialization after every time step, the narrow band method could not compete with the full domain version.
Parallelization is useful for both small and large images. Efficiency decreases when extra processors are added, but this decrease is smaller for large images than for small images. The MPI 1.1 does not support dynamic allocation of resources during the algorithm. The MPI 2.0 standard will support dynamic allocation of resources, which will make the multiresolution method more efficient; every time the algorithm moves from a coarse grid to a finer grid, more processors could be added.
Replacing synchronous communication functions with asynchronous ones made the DAS-2 [14] more efficient for most test cases. Efficiency did not change much for very large images. On the Cray T3E [13] that was used, no improvement in efficiency could be detected. Communication on the Cray is much faster than the DAS-2, but for the calculation time vice versa.
A possible future improvement could be adding more K-means optimizations to the algorithm. The average colors could be updated after every pixel is reassigned (on-the-fly K-means algorithm) or pixels can be randomly picked for reclassification instead of updating the full domain (R-means algorithm), however this makes parallelizing less efficient. Many initialization methods, like histogram based initialization, have been explored to improve the quality of the solution of the K-means algorithm or to decrease its calculation time and should be tested with the new segmentation algorithm.

5. REFERENCES

[1] D. Adalsteinsson & J.A. Sethian, A fast level set method for propagating interfaces, Journal of Computational Physics 118, 269 (1995)
[2] K.R. Castleman, Digital image processing (Prentice Hall, New Jersey, 1996)
[3] T.F. Chan, B.Y. Sandberg & L.A. Vese, Active contours without edges for vector-valued images, UCLA CAM report 99-35 (1999)
[4] T.F. Chan & L.A. Vese, Active contours without edges, UCLA CAM report 98-53 (1998)
[5] T.F. Chan & L.A. Vese, Variational image restoration & segmentation models and approximations, UCLA CAM report 97-47 (1997)
[6] M. Leeser, K-means algorithms for unsupervised classification, http://www.ece.neu.edu/groups/rpl/projects/kmeans/ (1999)
[7] S. Osher & R.P. Fedkiw, Level set methods, UCLA CAM report 00-08 (2000)
[8] J.R. Rommelse, High performance algorithms in image segmentation, MSc thesis, Delft University of Technology (2002)
[9] J.A. Sethian, Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision and materials science (Cambridge University Press, Cambridge, 1999)
[10] L.A. Vese & T.F. Chan, Image segmentation using level sets and the piecewise constant Mumford and Shah model, UCLA CAM report 00-14 (2000)
[11] L.A. Vese & T.F. Chan, Reduced non-convex functional approximations for image restoration & segmentation, UCLA CAM report 97-56 (1997)
[12] W.L. Wan, Scalable and multilevel iterative methods, UCLA CAM report 98-29 (1998)
[13] High performance applied computing, http://www.hpcn.tudelft.nl/ (2000)
[14] The distributed ASCI supercomputer 2 (DAS-2), http://www.cs.vu.nl/das2/ (2002)

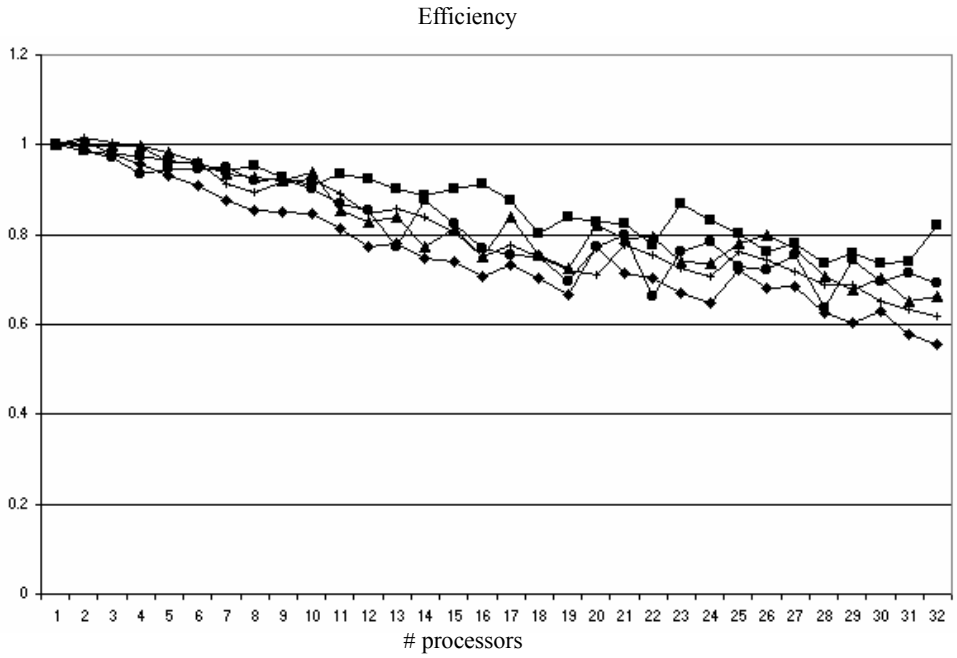


Figure 7 Efficiency with synchronous communication, measured on DAS-2, for several test cases.

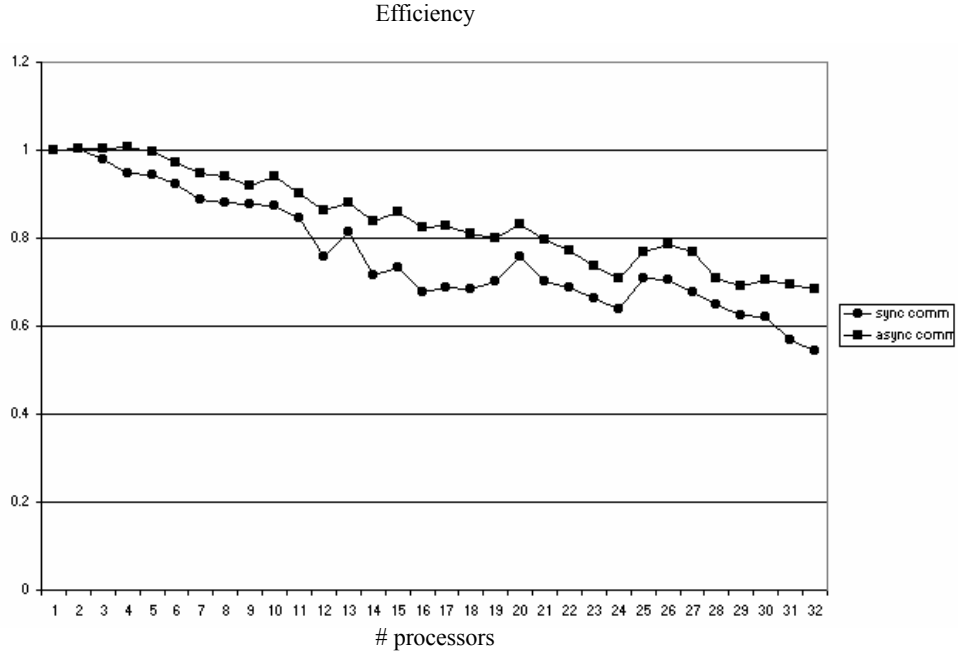


Figure 8 Efficiency improves if asynchronous communication is used on DAS-2, for some test cases.

An Improved Genetic Algorithm for Task Scheduling in Multi-Processor Environment

Man Lin
Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University
Antigonish, B2G 2W5, Nova Scotia, Canada
E-mail:mlin@stfx.ca

ABSTRACT

In real-time applications, task scheduling in multiprocessor architecture includes assigning tasks to processors and ordering the tasks in each processor while guaranteeing the timing constraints of the tasks. The problem of finding an optimal schedule for a set of tasks in a multi-processor environment is an NP-hard problem. Therefore, heuristic approaches are appropriate to these classes of problems. In this paper, we describe an improved Genetic Algorithm to solve the real-time scheduling optimization problems for tasks with deadline and ordering constraints. The improved Genetic Algorithm improves our previous work by enforcing the ordering constraint to all the individuals in a population. The evaluation process runs much faster since no repair has to be done. Furthermore, pre-processing for tasks takes into account the ordering constraints and thus eases the GA computation in assigning the starting time for tasks in a schedule. Experiment result shows that the improved GA finds better schedules in general.

Keywords: Genetic Algorithm, Scheduling optimization, Multi-processors

1. INTRODUCTION

With the advances in parallel processing technology, an application can be run efficiently in multiprocessor computing environment. Such an application can be decomposed to tasks which can be executed in different processors in parallel. Scheduling these tasks includes assigning tasks to processors and ordering the task in each processor while guaranteeing the dependency constraints between the tasks. In this paper, we consider real-time scheduling [1, 5] optimization problems for tasks with ordering constraints and deadline constraints which are commonly found in real-time systems. The aim of the scheduling optimization problem is to derive a schedule of a set of tasks on a set of processors which minimizes the completion time of these tasks while maintaining their temporal constraints. The possible constraints being considered are as follows:

- deadline constraints: task τ_i must finish before d_i,
- ordering constraints: task τ_j can only begin its execution after task τ_i finishes.

It is well known that such a scheduling optimization problem is an NP-hard problem [3]. Therefore, heuristic approaches are appropriate to these classes of problems. Various heuristic approaches such as Simulated Annealing, Tabu Search or Genetic Algorithms [2, 6] can be used. These heuristic techniques can avoid the problem of being trapped in local optima which is often seen in greedy search algorithms [2]. In [4], we used a Genetic Algorithm to solve the scheduling problem. However, due to the presence of infeasible solution, the evaluation of individual was complicated and the computation is slow because of the need to repairing infeasible solution. In this paper, we will present an improved Genetic Algorithm with pre-processing and feasibility control which eases the GA computation.

Basically, Genetic Algorithm maintains a *population* of individuals which represents a potential solution to the problem at hand. Each solution is evaluated to give some measures of its *fitness*. A new population is formed by selecting the more fit individuals. Some members of the new population undergo transformation by means of genetic *operators* including mutation and crossover to form new solutions. After some number of generations the program converges, it is hoped that the best individual represents a near-optimum solution. GAs have been applied in many optimization applications.

We use complex (non-string) representation and non-standard operators to the scheduling problem. As the tasks set has ordering constraints and deadline, if not properly designed, the individuals in a population will have a high chance of becoming infeasible. We will also discuss how topological order can help reducing the infeasible solution. Experimental results and comparison will be presented too.

The paper is organized as follows. We first describe the scheduling optimization problem in section2. Then we discuss the improved Genetic Algorithm in section 3. The experiment result is shown in section 4. Finally, the conclusion will be given.

2. PROBLEM DESCRIPTION

The scheduling problem consists of the following elements:

- A set P of processors $P_1, P_2, \ldots, Pn$. All processors are assumed to have the same performance.
- A set T of tasks $\tau_1, \tau_2, \ldots, \tau_m$ to execute. Each task τ_i has an execution time e_i, i.e. the time it takes to execute the task on any of the processors. Each processor can only execute one task at each time point. And the execution of a task is not preemptable, that is, it can not be interrupted or moved to another processor during execution.
- A *set O* of two types of temporal constraints.
  – Deadline constraint: $\tau_i < d_i$, where d_i is the deadline of τ_i.
  – Ordering Constraint: $\tau_i < \tau_j$ states that the execution of τ_i should end before the execution of τ_j can start, here τ_i is *provider* of τ_j and τ_j is *needer* of τ_i.

The following is an example of a task set with execution time and deadline.

τ_i	0	1	2	3	4	5
e_i	9	9	6	8	6	7
d_i	20	29	24	20	25	36

The following is an example of ordering constraint for the above task set. The ordering constraint is expressed as a set of (*provider, needer*)s. The first pair (2,3) in the table indicates that task τ_3 can only begin after τ_2 finishes.

provider	2	4	0	0
needer	3	5	5	1

Given a problem, our goal is to find the best schedule that has the least completion time while satisfying all the constraints. In order to do this, we need to know to which processor a task is assigned and the starting time of each task in each processor. We define s_i as the starting time for tasks τ_i and $Proc(i)$ as the processor number that task τ_i is executed. A solution to a scheduling problem is an assignment for each task $\tau_i \in T$ of a starting time *si* and a processor *Proc*(*i*) *2 P* such that:

- For each pair of tasks $\tau_i, \tau_j \in T$, if Pro(i) = Pro(i) then either $s_i+e_i \leqslant s_j$ or $s_j+e_j \leqslant s_i$(each processor can only execute one task at each time point).
- For each constraint $\tau_i < d_i$, it holds that $s_i+e_i < d_i$.

The aim is to minimize the end time of the last task, which is calculated as $max_1 \leqslant i \leqslant m(s_i + e_i)$.

3. THE GENETIC ALGORITHM

Genetic algorithms, well known for their robustness, are modern search techniques that start from an initial population of potential solutions to a problem, and gradually evolve towards better solutions through a repetitive application of genetic operators such as crossover and mutation. The evolution process proceeds through generations by allowing selected members of the current population, chosen on the basis of some criteria, to combine through a crossover operator to produce offspring thus forming a new population. The evolution process is repeated until certain criteria are met. We apply standard Genetic Algorithms to the scheduling optimization problem. As for any other domain-specific problem, we need to study

- how to represent solution in a population,
- how to evaluate a solution,
- how to generate the initial population,
- how to design the mutation and crossover operator,
- how to deal with infeasible solutions.

These questions will be answered in sequence in the following subsections.

Table 1 A Schedule

Task#	2	3	1	4	5
Proc#	1	2	2	1	1

3.1 Representation of a schedule

A tricky question is how to represent a schedule in a way suitable for a heuristic algorithm. The classical representation is binary strings, which is very restrictive and hard to understand from some applications point of view. We decided to use the following representation.

Task#	t_1	t_2	…	t_m
Proc#	p_1	p_2	…	p_m

where the first row is called *sche-row* which represents the schedule ordering of the tasks and the second row is called *proc-row* which represents the processor allocation of the corresponding tasks. A pair *t*, *p* in column *i* means that *sche*(*i*) = *t* and task τ_t will be executed on processor P_p, i.e. *Proc*(*t*) = P_p. Two tasks are explicitly ordered only if they are executed on the same processor.

Table 1 shows a schedule example. In this schedule, task τ_2 is allocated to processor P_1. Task τ_3 is allocated to processor P_2. Task τ1 is allocated to processor P_2 too. Since task τ_1 is behind task τ_3 in the first row, task τ_1 will be executed after τ_3 in processor P_2. For the same reason, processor P_1 executes task τ_2, τ_4 and τ_5 in sequence.

3.2 Evaluation

The aim of the scheduling problem is to minimize[1] the end time of the tasks. Given a schedule representation, we need to instantiate each task with a starting time. Given a schedule and processor assignment, there are many possible instantiations. The most common used policy is "as soon as possible". That is, assign the starting of a task to be the earliest possible time in the allocated processor. However, the earliest possible time for a task depends not only on the earliest available time of the processor, but also on the starting time of other tasks that must precede this task. Therefore, we need to perform preprocessing to the task set to calculate the earliest starting time of each task and then to derive an instantiation which takes into account the ordering constraints. The pre-processing and instantiation will be discussed the following sub-sections. The instantiated schedule may still be infeasible because the deadlines of some tasks are not met. In such case, we need to give some penalty to the schedule under consideration. In a schedule, if a task τ_i misses its deadline, then the evaluation adds a penalty of $s_i + e_i - d_i$ to this schedule. That is, a schedule is penalized by the total amount of time required beyond the deadline of those tasks that miss their deadline.

Table 2 A Task Set

τ_i	1	2	3	4	5
e_i	5	10	15	5	10
d_i	15	20	42	25	30

3.2.1 Pre-Processing

Based on the execution time of the tasks in the task set and the ordering constraints, we can derive the earliest starting time for each task. Let $prec(\tau_i)$ be the immediate predecessor of task *i* and $EST(\tau_i)$ be the earliest starting time of task τ_i. The earliest starting time for tasks without predecessors can be set to 0. And we have

$$EST(\tau_i) = \max_{j \in pree(\tau_i)} \{ EST(\tau_i) + e_j \}$$

Given a task set at table 2 and a ordering constraint set {τ2 < τ3}, the following shows the earliest starting time of the tasks.

EST(τ_i)	0	0	10	0	0

3.2.2 Instantiation

Given a schedule, the instantiation process iterates over the task in the first row (the sche-row) and assigns the starting time for task in sequence. For each task, it allocates the starting time in the assigned process (can be found at the proc-row) to allow it to start the earliest. That is, the starting time of a task is set to be the minimal value of the earliest starting time of the task and the earliest available time of the assigned processor. Note that the earliest starting time of a task depends on the starting time of the tasks preceding it.

[1] Note that in the implementation, we change the minimization problem to a maximization problem.

Given the task set at table 2 and the schedule at table 1, we can derive instantiate the schedule as follows:
Proc: 1 [T2: 0-10][T4: 10-15][T5: 15-25],
Proc: 2 [T3: 10-25][T1: 25-30].
We can check whether a schedule is feasible by examining the deadline constraints.

3.3 Initial population

In our previous algorithm [4], we generate the initial population for a scheduling problem as a number of random schedules. The problem was that these schedules can be infeasible because the tasks can have any order. Our previous algorithm then repairs the infeasible schedule by moving the tasks. The repair requires computation time. And it is possible that the derived schedule is still infeasible after a fixed number of steps of repair. In the worst case, the final best solution is an infeasible solution.
Our current algorithm makes use of topological order when generating the initial population. Instead of generating random schedule, we generate schedules that satisfies the partial ordering constraints. We use in-degree of a task T to indicate the number of tasks preceding T. The algorithm picks a task from the task set if there is no task preceding it. That is, if a task has in-degree 0 then it can be next in the topological order. We remove this task and look for another task of in-degree 0 in the resulting task set. We repeat until all task have been added to the topological order. If a topological order can not be found for a task set, then there will not be any feasible schedule for the scheduling problem under consideration. Any schedule bound to a topological order always satisfies the ordering constraint.
Suppose the following is ordering constraint for a task set with 6 tasks,

provider	0	5	2	2
needer	2	4	1	4

then the following is a topological order.

5	0	3	2	1	4

Note that deadline constraint may still be violated even a schedule has a topological order.

3.4 GA operators

GAs explore the search space by genetic operators. The mutation operator creates new individuals by a small change in a single individual. The crossover operator creates new individuals by combining parts from two individuals. The operators should be constructed to maintain one and only one occurrence of each task in the schedule. It is also preferable, but not necessary, that the ordering constraints are satisfied in the new individual. If an operation results in an illegal solution (the ordering constraints are violated), then a penalty will be imposed.
We choose the following mutation and crossover operators.

- The mutation operator either randomly change the processor of a randomly selected task.
- The crossover operator first randomly selecting a position. Child 1 use the ordering and processors of the tasks up to the selected position from parent 1, and the rest follows from parent 2 in order. The same mechanism applies to child 2.

Let's assume the following two individuals.
Parent 1:

Task#	2	3	1	4	0	··
Proc#	0	2	0	2	1	··

Parent 2:

Task#	0	4	3	2	1	··
Proc#	1	2	0	0	0	··

Suppose the selected position is 2, then the new individuals will be:
Child 1:

Task#	2	3	0	4	1	··
Proc#	0	2	1	2	0	··

Child 2:

Task#	0	4	2	3	1	··
Proc#	1	2	0	2	0	··

The crossover operator will produce two offsprings that are topological ordered with respect to the partial ordering if the parents have topological orders. This is proved in [7].

3.5 Infeasible solutions

Topological ordering enforces the scheduling to meet the ordering constraints. However, the deadline constraint can not be guaranteed. Therefore, we may have infeasible solution in a population. The question is, shall we replace a parent with the child which is illegal or keep the individual unchanged? Our method is to *replace* it according to the probability. Thus, in one generation, the population might contains both legal and illegal solutions, which provides the possibility to explore the whole search space. The replacement probability is often very low, therefore, we can keep most of the individuals in one population legal.

4. EXPERIMENTS

We performed empirical tests over a large number of scheduling problems. The problems were randomly generated. They varied over the size of task set, the execution duration of each task in the task set, the number of processors, the constraints of both types. The following shows the comparison of using randomly generated individuals (R-type algorithm) and individuals bound to topological order (T-type algorithm) for some scheduling problems. We use R-end and T-end to denote the ending time of the best solution for R-type and T-type algorithm, respectively. The end time with (IN) denotes the final solution is an invalid solution.
In general, T-type finds better solution, especially for problems with tight constraints.

N_t	N_p	N_c	R-end	T-end
10	4	25	41	40
10	5	15	37	30
20	3	15	49	49
20	5	20	37	31
30	3	15	77	75
30	3	20	78	76
30	3	25	78	77
30	4	15	59	66
30	4	20	65	62
40	3	15	108(*IN*)	99
40	3	20	103	110
40	3	25	102(*IN*)	113
40	4	15	80	78
40	4	25	80	76
50	3	15	128	148
50	3	20	125	125
50	3	25	151(*IN*)	125
50	4	20	106	97
50	4	25	103	99
100	5	15	154	153

100	5	20	158	153
100	5	30	162	154(*IN*)
100	5	35	153(*IN*)	152
100	5	40	158(*IN*)	155

5. CONCLUSION

One strength of GAs is their robustness, which is mainly caused by the fact that they deal with a sample of candidate solutions to an optimization problem at a time. In this paper, we have described an improved Genetic Algorithm to solve real-time scheduling optimization problem in a multi-processor architecture. First of all, we add deadline constraints to tasks. This makes our scheduling problem more general. Second, we improve the GA algorithm by imposing ordering in any individual in a population and performing preprocessing before assigning starting time of tasks, which decreases the chances of having infeasible solution to the scheduling problem. The experimental results have also been shown.

6. ACKNOWLEDGMENTS

The author would like to thank NSERC (National Science Engineering Research Council, Canada) for the support of this research.

7. REFERENCES

[1] A. Burns and A.Wellings. *Advances in Real-Time Systems*, chapter 3. Prentice Hall Inc., 1995.

[2] F. Glover and M. Laguna. *Modern Heuristic Techniques for Combinatorial Problems*, chapter Tabu search. Blackwell Scientific Publications, 1993.

[3] S-T. Levi and A. Agrawala. *Real-Time Systems Design*. McGraw-Hill, 1990.

[4] M. Lin and L.T. Yang. Hybrid genetic algorithms for partially ordered tasks in a multi-processor environment. In *Proceedings of 6th Int'l Conf. On Real-Time Comp. Systems and App.*, pages 382–387, 1999.

[5] J. Liu and R. Sha. *Advances in Real-Time Systems*, chapter 9. McGraw-Hill, 1990.

[6] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, second edition, 1994.

[7] Jaewon Oh and Chisu Wu. Genetic-algorithmbased real-time task scheudling with multiple goals. draft, 2002.

Hierarchical Parallel Algorithms for Module Placement of Large Chips on Distributed Memory Architectures*

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

ABSTRACT

The PROUD module placement algorithm mainly uses a hierarchical decomposition technique and the solution of sparse linear systems based on a resistive network analogy. It has been shown that the PROUD algorithm can achieve a comparable design of the placement problems for very large circuits with the best placement algorithm based on simulated annealing, but with several order of magnitude faster. The modified PROUD, namely MPROUD algorithm by perturbing the coefficient matrices performs much faster that the original PROUD algorithm. Due to the instability and unguaranteed convergence of MPROUD algorithm, we have proposed a new convergent and numerically stable PROUD, denoted as SPROUD, with attractive computational costs to solve the module placement problems by making use of the SYMMLQ method based on Lanczos process in [16]. In this paper, we subsequently propose a parallel version of the SPROUD algorithm. The parallel algorithm is derived such that all inner products and matrix-vector multiplications of a single iteration step are independent. Therefore, the cost of global communication which represents the bottleneck of the parallel performance on parallel distributed memory computers can be significantly reduced, therefore, to obtain another order of magnitude improvement in the runtime without loss of the quality of the layout.

Keywords: - Chip module placement, large and sparse linear systems with symmetric indefinite coefficients, parallel systems, the SPROUD algorithm, Lanczos process, the SYMMLQ method

1. INTRODUCTION

The main objective of the module placement problem is to place a set of modules such as standard cells, gate arrays, and sea-of-gates on a very large and highly complex chip and minimize the total wire-length between the modules, given a special netlist which addresses the connectivity between the modules. Our attention is mainly focus on the global placement based on two assumptions that all modules are point module, i.e. are of equal size and the connections to the nets are assumed to be at the center of the module and all nets are two-pin nets. Multi-pin nets are processed and replaced with two-pin nets.

Originally, the method used the quadratic placement formulation first proposed by Hall [6]. Since it is quite hard to find the eigenvalues and eigenvectors of a large and sparse matrix. Tsay, Kuh and Hsu [11] introduce a hierarchical placement method, PROUD, which takes full sparsity inherent in the placement specification. Their approach takes the I/O pad specification as input and successively solves sparse linear system of equations. It depends on the concept of resistive network optimization proposed by Cheng and Kuh [1], but bypasses the slot constraint and employs a simpler partitioning scheme. Then the block Gauss-Seidel iteration to resolve the placement interactions among different blocks after partitioning. The PROUD algorithm is especially suitable for extremely large design having tens of thousands of gates and produces placement qualities comparable to those obtained by simulated annealing [9] but in runtimes that are orders of magnitude faster.

In [12], Xing and Banerjee have proposed a modification on PROUD, namely MPROUD to the connection matrix based on the concept of perturbation. They have shown experimentally how this MPROUD algorithm improves the runtimes by an order of magnitude without loss of quality of the layout. They also subsequently parallelize the original and modified module placement algorithms using Message Passing Interface (MPI) on various multiprocessor systems and report the experimental results of the algorithms on a set of large layout benchmarks. Due to the instability and unguaranteed convergence of MPROUD algorithm where for the symmetric indefinite coefficient matrix, Gauss-Seidel method is used with perturbation to prevent the singularity, we have proposed two new convergent and numerically stable PROUDs, namely SPROUD and IPROUD algorithms [16, 17] by making use of the SYMMLQ and MINRES methods, respectively, proposed by Paige and Saunders [8] with attractive computational costs to solve the module placement problems. Experimental results produced by our approach are reported for a variety of large layout benchmark circuits that the SPROUD algorithm is very suitable for large designs and produces placement qualities comparable to those obtained by simulated annealing [9] but in runtimes that are orders of magnitude faster. Meanwhile it is also a fast, stable and efficient algorithm.

In this paper, we mainly concern the parallel design for this PROUD-based algorithm. Previously the parallel version of IPROUD has been described in [13]. Here we mainly propose a parallel version of the SPROUD algorithm. The parallel algorithm is derived such that all inner products and matrix-vector multiplications of a single iteration step are independent. Therefore, the cost of global communication which represents the bottleneck of the parallel performance on parallel distributed memory computers can be significantly reduced, therefore, to obtain another order of magnitude improvement in the runtime without loss of the quality of the layout.

The paper is organized as follows. Section 2 describes briefly the PROUD and MPROUD algorithms. Then our fast, numerical stable and convergent algorithm SPROUD is presented fully in section 3. Then in section 4, we describe the basic idea of the parallel hierarchical algorithms suitable for various multiprocessor systems such as the Intel Paragon, the Parsytec, the SGI Challenge, CrayT3E, and a cluster of workstations. Finally, section 5 describes some limited preliminary experimental results produced by the parallel version of SPROUD algorithm compared with the corresponding PROUD and MPROUD algorithms for a variety of large layout benchmark circuits.

2. THE PROUD AND MPROUD ALGORITHMS

In this section, we mainly review the PROUD [10,11] and MPROUD [12] algorithms, respectively. The PROUD is a

hierarchical algorithm. It has two major components, namely global placement by solving a system of equations, and partitioning based on the results of placement. Global placement determines the relative positions of the modules in the current block for the partitioning routine. By sorting the modules by positions, the partitioning stage bisects the current block into two sub-blocks. Global placement is mainly based on two assumptions, namely, that all modules are point modules and that all nets are two-pin nets. Thus multi-pin nets are expanded into two-pin net cliques.

The weight of each edge of an r-dimensional net is set to 2/r. Let matrix $C = (C_{jj})$, where $C_{ii} = 0$ and Cij be the number of nets connecting module i and module j. Let (x_i, y_i) represent the location of the module i and $\bar{x}$ and $\bar{y}$ represent the n_dimensional vectors which specify the coordinates of n modules. The objective is

to minimize the sum of the squared wire lengths

$$L(\bar{x},\bar{y}) = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij}[(x_i - x_j)^2 + (y_i - y_j)^2] \quad (1)$$

Let D be diagonal matrix with entries $d_{ii} = \sum_{y=1}^{n} c_{ii}$

and B = D - C, then

$$L(\bar{x},\bar{y}) = \bar{x}^T B\bar{x} + \bar{y}^T B\bar{y} \quad (2)$$

Based on a resistive network analogy, the minimization of the term $\bar{x}^T B\bar{x}$ can be achieved by solving the following system of equations:

$$B_{11}\bar{x}_1 + B_{12}\bar{x}_2 = 0 \quad (3)$$

where $\bar{x}_1$ is the vector of x coordinates all movable modules in current block and $\bar{x}_2$ is the vector of x coordinates all pads and all other modules not inside current block. Here we usually assume that all I/O

pad locations are fixed. B_{11} consists of columns of B with the same indices as $\bar{x}_1$. Since B is symmetric, B_{11} is also symmetric. Correspondingly B12 consists of columns of B with the same indices as $\bar{x}_2$. At each level of the hierarchy, there is a coordinate used by global placement. Supposed the current coordinate is x. Global placement obtains the relative positions,

over x coordinate, the chip area is partitioned into two equal-sized sub-blocks with roughly equal number of modules in each sub-block. Then the new blocks will placed by global placement. The loop of global placement and partitioning is executed a few time to make the placement more accurate. This number is called block Gauss-Seidel number [10]. In next level of the hierarchy, the coordinate is switched to y. This process continues until each block contains at most one module. In the MPROUD algorithm, Xing and Banerjee [12] make some modifications to (3) where let $\bar{b} = -B_{12}\bar{x}_2$, then equation (3) becomes

$$B_{11}\bar{x}_1 = \bar{b} \quad (4)$$

It is quite clear that the coefficient matrix B11 is always symmetric. The convergence rate by Gauss-Jacobi or Guass-Seidel algorithms for solving a linear system of equations like (4) is mainly determined by the condition number of B_{11}. When B_{11} tends to be singular, the condition number increases, and correspondingly the convergence rate decreases. Then they propose to proceed as follows:

Definition 2.1 Let $B_{11} = D - C$, where $D = d_{ii}$ is sub-matrix of B_{11} with diagonal entries and $C = (c_{ij}), i \neq j$ sub-matrix of B_{11} with o_-diagonal entries. We call $B'_{11} = (1+\varepsilon)D - C$ the diagonally perturbed matrix of B_{11}. If we use B'_{11} for B_{11} in (3), the algorithm is called MPROUD.

The nice property of MPROUD is that the matrix B_{11} is always non-singular, and the condition number is always smaller than that in PROUD. Therefore it can achieve a good improvement in performance.

3. THE SPROUD ALGORITHM

However, due to the limitation on convergence of Gauss-Seidel algorithm which can not ensure the corresponding splitting to be regular [5], relatively expensive computational costs and unstable numerical properties, we can not consider it to be the efficient iterative method for solving the large and sparse linear system (4). In this section, we make use of the stable and computationally attractive SYMMLQ method [8] based on Lanczos process to solve the large and sparse linear system (4) whose coefficient matrix is symmetric, but possibly indefinite. Additionally, the IPROUD algorithm is also described briefly as well.

The Lanczos process [7] which is very powerful and widely used in many applications is closely related to the conjugate gradient method for solving symmetric linear systems. It can be derived as a method for reducing the symmetric matrix B11 to tri-diagonal form

$$T = V^T B_{11} V,$$

where $V = v_1, v_2, \cdots, v_n$ is orthogonal, and

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_j \\ & & & \beta_j & \alpha_j \end{pmatrix}$$

Equating the first n-1 columns in $B_{11} = (v_1, v_2, \cdots, v_n)$

$= (v_1, v_2, \cdots, v_n)T$ gives

$$\beta_j v_j - 1 = B_{11} v_j - \alpha_j v_j - \beta_{j+1} v_{j+1} \quad (5)$$

where $j = 1, \cdots, n-1$ and we have put $\beta_1 v_0 \equiv 0$.

Multiplying this equation by v_j^T and using that $v_j \perp v_{j+1}$ we get

$$\alpha_j = v_j^T (B_{11} v_j - \beta_j v_{j-1})$$

Further solving for v_{j+1},

$$\beta_{j+1} v_{j+1} = r_{j+1}, r_{j+1} = B_{11} v_j - \alpha_j v_j - \beta_j v_{j-1},$$

so if $r_{j+1} \neq 0$, then β_{j+1} and v_{j+1} are obtained by normalizing r_{j+1}.

In the context of solving the large and sparse linear system (4), the appropriate choice is

$\beta_1 v_1 = \bar{b}, \beta_1 = \left\|\bar{b}\right\|_2$.

It is easy to see that if the Lanczos process can be carried out for k steps then it holds

$$B_{11}V_k = V_k T_k + \beta_{k+1} v_{k+1} e_k^T, \quad (6)$$

and span $(V_k) = K_k(B_{11}, \bar{b})$ where Krylov subspace $K_k(B_{11}, \bar{b}$ is de_ned as

$K_k(B_{11}, \bar{b}) = \text{span}\{\bar{b}, B_{11}\bar{b}, \cdots, B_{11}^{k-1}\bar{b}\}$

MINRES method seeks an approximation solution of (4) as follows:

$$\left\|\bar{b} - B_{11}\bar{x}_1\right\|_2 = \min_{x \in S_k} \left\|\bar{b} - B_{11}x\right\|_2,$$

and

$S_k = x_0 + K_k(B_{11}, r_0),$

where $r_0 = \bar{b} - B_{11}x_0$.

In term of the Lanczos iteration the approximation solutions $x_k = V_k y_k$ are determined by

$$V_k^T B_{11}^2 V_k y_k = V_k^T B_{11}\bar{b} \quad (7)$$

From (6) we see that

$$V_k^T B_{11}^2 V_k = T_k^2 + \beta_{k+1}^2 e_k e_k^T, \quad (8)$$

$$V_j^T B_{11}\bar{b} = \beta_1 V_k^T B_{11} v_1 = \beta_1 T_k e_1. \quad (9)$$

If we carry out the orthogonal factorization described fully in [8] we see that

$$T_k^2 + \beta_{k+1}^2 e_k e_k^T = \bar{L}_k \bar{L}_k^T + \beta_{k+1}^2 e_k e_k^T = L_k L_k^T \quad (10)$$

Then we have to solve the following equation

$$L_k L_k^T y_k = \beta_1 \bar{L}_k Q_k e_1 \quad (11)$$

Because $\bar{L} = L_k D_k$ and $D_k = \text{diag}(1,2,\cdots,1,c_k)$, this simplifies to

$$L_k^T y_k = \beta_1 D_k Q_k e_1 \quad (12)$$

If we write $T_k = V_k y_k = M_k t_k$, where $M_k = V_k L_k^{-T}$ and $t_k = L_K^T y_k$, then we can again update M_k and t_k as k increase.

SYMMLQ method seeks an approximation solution for $\bar{x}_k$ by $T_k = V_k y_k$. The stationary values are given by the Galerkin condition:

$V_k^T(\bar{b} - B_{11}V_k y_k) = 0.$

Multiplying (6) by V_k^T *k* , we obtain the tridiagonal system

$$T_k y_k = \beta_1 e_1 \quad (13)$$

To solve the tridiagonal system Paige and Saunders suggest computing the orthogonal factorization

$$T_m = \bar{L}_k Q_k, Q_k^T Q_k = I,$$

where $\bar{L}_k$ is lower triangular. Such a factorization always exists, and can be computed by multiplying T_k with a sequence of plane rotations from the right

$$T_k G_{12} \cdots G_{k-1,k} = \bar{L}_k = \begin{pmatrix} \gamma_1 & & & & \\ \delta_2 & \gamma_2 & & & \\ \varepsilon_3 & \delta_2 & \gamma_2 & & \\ & \ddots & \ddots & \ddots & \\ & & \varepsilon_k & \delta_k & \bar{\gamma}_k \end{pmatrix}$$

The rotation $G_{k-1,k}$ is defined by elements c_{k-1} and S_{k_1}. The bar on the elements $\bar{\gamma}_k$ is used to indicate that $\bar{L}_k$ differs from L_k, the *k* k* leading part of $\bar{L}_{k+1}$, in the (*k, k*) elements only. In the next step the elements in $G_{k,k+1}$ are given by

$$\gamma_k = (\bar{\gamma}_k^{\,2} + \beta_{k+1}^2)^{1/2}, c_k = \bar{\gamma}_k/\gamma_k, S_k = \beta_{k+1}/\gamma_k,$$

Since y_k will change fully with each increase in k we write

$$V_k y_k = (V_k Q_k^T)\bar{x}_k = \bar{W}_k \bar{x}_k,$$

where

$$\bar{W}_k = (w_1, \cdots, w_{k-1}, \bar{w}_k),$$

and

$$\bar{x}_k = (\varsigma_1, \cdots, \varsigma_{k-1}, \bar{\varsigma}_k) = Q_k y_k$$

Here quantities without bars will be unchanged when k increases, and $\bar{W}_k$ can be updated with $\bar{T}_k$

The system now becomes

$$\bar{L}_k z_k = \beta_1 e_1, \quad x_k^c = \bar{W}_k z_k$$

The formulation allows the v_i and ω_i to be formed and discarded one by one. In implementing the algorithm we should note that x_k^c need not be updated at each step, and that if $\bar{\gamma}_k = 0$, then x_k is not defined. Instead we update

$$x_k^L = W_k z_k = x_{k-1}^L + \varsigma_k w_k$$

where L_k is used rather than $\bar{L}_k$. We can then always obtain x_{k+1}^c when needsssed from

$$x_{k+1}^c = x_k^L + \bar{\varsigma}_k^L + \bar{\varsigma}_{k+1}\bar{w}_{k+1}$$

This completely defines the SYMMLQ algorithm. The practical details of implementation are discussed at length in [8]. Most of the potential ill-conditioning or singularity resulting from the coefficient matrix B11 has been avoided. This is a numerically stable and fast algorithm with attractive computational costs. It should be noted that MINRES su_ers more from poorly conditioned systems than SYMMLQ does. That is why we can see the proposed SPROUD based on SYMMLQ achieves better speedup and smaller degradation than IPROUD based on MINRES.

4. PARALLEL SPROUD ALGORITHM

On massively parallel computers, the basic timeconsuming computational kernels of the SPROUD algorithm are usually: inner products, vector updates,matrix-vector multiplications. In many situations, especially when matrix operations are well-structured, these operations are suitable for implementation on vector and share memory parallel computers [4]. 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 e_ciently by parallel operations, we still can not avoid the global communication, i.e. communication of 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 [2, 3]. These global communication costs become relatively more and more important when the number of the parallel processors is increased and thus they have the potential to affect the scalability of the algorithm in a negative way [2, 3].

For proposed parallel SPROUD algorithm, the basic idea is to reorganize the algorithm such that all inner products and matrix vector multiplication of a single iteration step are independent [18, 14, 15]. Therefore, the cost of global communication which represents the bottleneck of the parallel performance on parallel distributed memory computers can be significantly reduced, therefore, to obtain another order of magnitude improvement in the runtime without loss of the quality of the layout. Due to the space limitation, the detailed algorithm will be described later.

After getting the parallel algorithm, the parallel implementation can illustrated on a chip by the following example proposed in [12]. Suppose we have 16 modules to be placed and there are 4 processors available. At the first level of the hierarchy, all 4 processors participate in the global placement. A master process will control which variables a slave processor needs to iterate when concurrently working on our proposed iteration. After placement the master processor bisects the portion of the chip it own into two sub-chips with roughly equal number of modules on each. At the same time the working is partitioned into two. Processor p0 and p1 will jointly place 8 modules and p2 and p3 will place the remaining 8 modules. Both 2 processor groups work exactly the same way as the first hierarchy. After one more level each working group will have only 1 processor. At the same time this processor will execute the sequential SPROUD algorithm.

5. EXPERIMENTAL RESULTS

In this section, we mainly present the experimental schemes to be carried out on a variety of large MCNC layout benchmark circuits using MPI, a standard specification for message-passing libraries, on various multiprocessor systems such as the Intel Paragon, the Parsytec, the SGI Challenge, CrayT3E, and a cluster of Sparc10 workstations.First we choose benchmark circuits Industry-1,Industry-2, Industry-3 from the MCNC layout synthesis benchmarks as our test circuits. The corresponding characteristics from these benchmark circuits such as cell, pad and net numbers. Comparing our new approach, parallel implementations of SPROUD with parallel version of PROUD, MPROUD and IPROUD algorithms respectively, we will consider the overall parallel performances and qualities of the layouts measured in terms of total wire length.On a comparison, on these parallel architectures,we run TimberWolf6.0 based on simulated annealing, parallel versions of original PROUD, MPROUD and IPROUD algorithms with _ = 0:1 and our approach,namely SPROUD, for Industry-1, Industry-2, Industry3, assuming that all circuits have equal size modules and all nets have two terminals. We _rst run those algorithms and our SPROUD approach, then input the placement of the SPROUD to TimberWolf6.0 to evaluate the cost of placement. The comparison based on some limited preliminary experimental results in term of runtimes and speedup is quite promising. Further extensive measured results conducted will be reported shortly in the future.

6. REFERENCES

[1] C.K. Cheng and E. S. Kuh. Module placement based on resistive network optimization. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 3:218{225, 1984.

[2] 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.

[3] 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.

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

[5] G. H. Golub and C. F. Van Loan. Matrix Computations. Johns Hopkins University Press, Baltimore, 2nd edition, 1989.

[6] K. Hall. An r-dimensional quadratic placement algorithm. Management Science, 17(3):219{229, November 1970.

[7] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. Journal of Research of National Bureau of Standards, 45:255{282, 1950.

[8] C. C. Paige and M. A. Saunders. Solution of sparse inde_nite systems of linear equations. SIAM Journal on Numerical Analysis, 12(4):617{629, 1975.

[9] C. Sechen and A. Sangivovanni-Vincentelli. The TimberWolf placement and routing package. IEEE Journal for Solid State Circuits, SC-20:510{522, 1985.

[10] R-S. Tsay and E. Kuh. A fast sea-of-gates placement algorithm. In Proceedings of 25th ACM/IEEE Design Automation Conference, pages 318{322, 1988.

[11] R-S. Tsay, E. Kuh, and C-P. Hsu. Module placement for large chips based on sparse linear equations. International Journal of Circuit Theory and Applications, 16:411{423, 1988.

[12] Z. Xing and P. Banerjee. A parallel hierarchical algorithm for module placement based on sparse linear equations. In Proceedings of the 1996 International Conference on Circuits and Systems, Atlanta, GA, May 1996.

[13] L. T. Yang. Parallel efficient hierarchical algorithms for module placement of large chips on distributed memory architectures. In Proceedings of the International Conference on Parallel Computing in Electrical Engineering (ParElec-02), Warsaw, Poland, September 22-25, 2002.

[14] L. T. Yang and R. P. Brent. The improved biconjugate gradient method on parallel distributed memory architectures. In Workshop Proceedings of the 16th International Parallel and Distributed Processing Symposium(IPDPS-PDSECA02), pages 233{240, Fort Lauderdale,Florida, April 15-19, 2002.

[15] L. T. Yang and R. P. Brent. Quantitative performance analysis of the improved quasi-minimal residual method on massively distributed memory computers. Advances in Engineering Software, 33:169{177, 2002.

[16] T. Yang. An efficient and stable hierarchical algorithm based on sparse linear system for module placement. In Proceedings of the 8th IEEE International

Workshop on Circuits, Systems and Signal Processing (CSSP-97), pages 701{708, 1997. November 27-28, Mierlo, The Netherlands.

[17] T. Yang. An efficient hierarchical algorithm for module placement for large chips. In Proceedings of the 7th International Symposium on Integrated Circuits, Technology, Systems and Applications (ISIC-97), pages 319{322, September 10-12, 1997. Singapore.[18] T. Yang and H. X. Lin. The improved quasi-minimal residual method on massively parallel distributed memory computers. IEICE Transactions on Information and Systems, E80-D(9):919{924, September 1997. Special issue on Architectures, Algorithms and Networks for Massively Parallel Computing.

The Parallel Quantum Simulator and its Application on Prime Factorization *

Xiao Youan And Li Layuan
Information Insistent, Wuhan University of Technology
Wuhan, Hubei, 430063, P.R.China
E-mail: youan@public.wh.hb.cn

ABSTRACT

Quantum Computer is a new type of computer which can efficiently solve complex problems such as prime factorization. Because of the exponential nature of quantum computers, simulating the effect of errors on them requires a vast amount of processing and memory resources. In this paper, a parallel simulator is described and the application on prime factorization is also introduced.

Keywords: Parallel Simulator, Quantum Computer, prime factorization

1. INTRODUCTION

A quantum computer consists of atomic particles which obey the laws of quantum mechanics. The complexity of a quantum system is exponential with respect to the number of particles. Performing computation using these quantum particles results in an exponential amount of calculation in a polynomial amount of space and time. This quantum parallelism is only applicable in a limited domain. Prime factorization is one such problem which can make effective use of quantum parallelism. This is an important problem because the security of the RSA public-key cryptosystem relies on the fact that prime factorization is computationally difficult.

Errors limit the effectiveness of any physical realization of a quantum computer. These errors can accumulate over time and render the calculation useless. The simulation of quantum circuits is a useful tool for studying the feasibility of quantum computers. Simulations inject errors at each step of the calculation and can track their accumulation.

Because of the exponential behavior of quantum systems, simulating them on conventional computers requires an exponential amount of operations and storage. For this reason, to simulate problems of interesting size the use of parallel supercomputers is needed. In this paper we describe a parallel simulator algorithm which allows the simulation of circuits which are three to four orders of magnitude larger than any current proposed experimental realizations of a quantum computer.

The remainder of this section gives a brief introduction to quantum computers. In section 2 we describe the parallel simulation algorithm methodology and in section 3 we introduce the application on Prime factorization. In section 4 we conclude.

1.1 Quantum Computers

A quantum computer performs operations on bits, called QU-bits, whose values can take on the value of one or zero or a superposition of one and zero. This superposition allows the representation of an exponential number of states using a polynomial number of QU-bits. A quantum computer performs transformations on these QU-bits to implement logic gates. Combinations of these logic gates define quantum circuits.

1.2 QU-bits and Quantum Superposition

The basic unit of storage in a Quantum Computer is the QU-bit. A QU-bit is like a classical bit in that it can be in two states, zero or one. The QU-bit differs from the classical bit in that, because of the properties of quantum mechanics, it can be in both these states simultaneously. A QU-bit which contains both the zero and one values is said to be in the superposition of the zero and one states. The superposition state persists until we perform an external measurement. This measurement operation forces the state to one of the two values. Because the measurement determines without doubt the value of the QU-bit, we must describe the possible states which exist before the measurement in terms of their probability of occurrence. These QU-bit probabilities must always sum to one because they represent all possible values for the QU-bit.

Bit Value	Amplitude	Probability
0	1	1
1	0	0

(a) Representation of a 0 bit value

Bit Value	Amplitude	Probability
0	$1/\sqrt{2}$	1/2
1	$1/\sqrt{2}$	1/2

(b) Representation of a 0 bit value

Figure 1 Vector representation of QU-bit values

The quantum simulator represents a QU-bit using a state vector. Figure 1 shows how the simulator uses complex amplitudes to represent a QU-bit. Each state in the vector represents one of the possible values for the QU-bit. The bit value of a state corresponds to the index of that state in the vector. The simulator represents each encoded bit value with a non zero amplitude in the state vector.

The probability of each state is defined as the square of this complex amplitude. Figure 1(a) shows a state which represents the single value of zero. In Figure 1(b) the probability is equally split between the zero and one states, representing a QU-bit which is in the superposition state. For a register with M QU-bits, the simulator uses a vector with 2^M states.

An M QU-bit register can represent 2^M simultaneous values by putting each of the bits into the superposition state. A calculation using this register calculates all possible outcomes for the 2^M input values, thereby giving exponential parallelism. The bad news is that in order to read out the results of a calculation we have to observe, i.e. measure, the output. This measurement forces all the bits to a particular value thereby destroying the parallel state. The challenge then is to devise a quantum calculation where we can accumulate the parallel state in non-exponential time before performing the measurement.

* This work is supported by NSF of China.

1.3 Quantum Transformations and Logic Gates

A quantum computation is a sequence of transformations performed on the QU-bits contained in quantum registers. A transformation takes an input quantum state and produces a modified output quantum state. Typically we define transformations at the gate level, i.e. transformations which perform logic functions. The simulator performs each transformation by multiplying the 2^M dimensional vector by a 2^M x 2^M transformation matrix.

The basic gate used in quantum computation is the controlled-not, i.e. exclusive or gate. The controlled-not gate is a two bit operation between a control bit and a resultant bit. The operation of the gate leaves the control bit unchanged, but conditionally flips the resultant bit based on the value of the control bit. Table 1 shows a truth table of how the controlled-not gate modifies the different QU-bit values. In the vector representation of the QU-bits, the controlled-not gate corresponds to a transformation which swaps the amplitude of the states in the third and fourth positions. Figure 2 shows the 4 x 4 matrix which performs the controlled-not transformation on the two QU-bits.

Table 1 Truth Table foe the Control-not gate

Input Bits		Output Bits	
A	**B**	**A′**	**B′**
0	**0**	**0**	**0**
0	**1**	**0**	**1**
1	**0**	**1**	**1**
1	**1**	**1**	**0**

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \times \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ a_3 \\ a_2 \end{pmatrix}$$

Swap Amplitude a_2 and a_3

Figure 2 Controlled not Transformation

1.4 The Physical Realization of a Quantum Computer

The ion trap quantum computer, proposed by Cirac and Zoller, is one of the most promising schemes for the experimental realization of a quantum computer. Other promising schemes include cavity QED and Quantum dots. To date, cavities trapping up to 33 QU-bits have been constructed, and simple quantum gates have been demonstrated. We model our simulations directly on the Cirac and Zoller scheme. The ion trap computer uses the internal energy states of the ions to encode the zero and one states of a QU-bit. This scheme also requires a third temporary state to implement gates such as the controlled-not gate. Lasers directed at the individual ions define a series of transformations which when combined implement the controlled-not gate.

In general the physical process used to perform the transformations will not be performed perfectly. The resulting inaccuracies, referred to as operational errors, degrade the calculation over time. Also interaction of the QU-bits with the external environment has a destructive effect on the coherence of the superposition state. This type of error, referred to as decoherence, acts to measure the QU-bits and thereby destroys the parallel state. Both operational and decoherence errors act to limit the effectiveness of a quantum computer. Simulation is an effective tool for characterizing errors, and tracking their accumulation. Using a physical model such as the Cirac and Zoller scheme is important for obtaining realistic results.

1.5 Quantum Algorithms

Much of the current interest in quantum computation is due to the discovery of an efficient algorithm to factor numbers. This is an important problem because a quantum factoring engine would severely threaten the security of public-key cryptosystems. The quantum factoring algorithm uses quantum parallelism to calculate all of the values of a function simultaneously. This function is periodic and a quantum FFT can extract this period efficiently. We then use a polynomial time classical algorithm to calculate the factors from this period.

Quantum computers are also useful for searching unsorted databases. The quantum search algorithm runs in time for items, where the best classical algorithm runs in time. Therefore for NP-complete problems such as circuit SAT, which contains items for m variables, the quantum algorithm runs in $O(2^m/2)$ time. This speedup is not exponential like that of the factoring algorithm, but it allows the solution of problems that may be computationally intractable on a classical computer.

2. PARALLEL QUANTUM SIMULATION ALGORITHM

Because of the exponential nature of quantum computing, simulation of it on a conventional computer requires exponential memory and processing resources. For a medium sized problem, like a single modulo multiplication step from the Factor 15 problem, we need to represent of 3^{16} = 43,046,721 states if we use the detailed three state model. This translates to over 600 Megabytes of storage. There are also about 8000 operations in the simulation, and each operation must operate on each of the states. Fortunately we can split up the calculation and use the memory and processing resources of multiple processors.

2.1 Parallel Division of a Quantum Computer Simulation

Quantum simulation involves a series of independent transformations on the vector space used to represent the QU-bits. Because of this inherent parallelism, quantum computer simulation is a natural candidate for parallel processing. For an M QU-bit quantum computer, the vector space for the detailed Cirac and Zoller model is of size $V=3^M$. A single transformation on this state space is therefore a sequence of 3^M-1 simple operations which can all be run in parallel. For a circuit with 16 QU-bits, this represents a maximum degree of parallelism of 3^{15}.

One potential obstacle to parallelizing a quantum computation is the reorganization of data that may be needed between the simulation of successive gates. The bits used by a gate define how the simulator divides the vector space into two element vectors. To exploit all the available parallelism, the simulator must partition all the two element vectors to separate processors. Because there is excess parallelism, the simulator can allocate multiple two element vectors on each processor. The allocation needs to change only when we need to perform a transformation which is not covered by the original allocation.

To exploit a degree of parallelism of 3^N, the simulator picks a set of N QU-bits to parallelize across. To determine the allocation of the QU-bit states, the simulator concatenates the N QU-bit values of each state and allocates that state to the processor whose ID is equal to the concatenated value. For a quantum simulation of M QU-bits, each of the processors will have 3^{M-N} states allocated on it. All of the processors can run in parallel operating only on local data as long as they do not use any of the QU-bits in the set N. When the computation

needs to operate on one of the bits in N, a new set is picked and the simulator redistributes the data. The entire simulation is a sequence of these computation and reorganization steps. The efficiency of the parallel simulator now becomes a question of how often the simulator needs to redistribute the data.

$$f(A) = X^{A} \bmod N = X^{[a_0 2^0 + a_1 2^1 \ldots + a_l 2^l]} \bmod N \quad \text{(Eq 1)}$$
$$= X^{a_0} X^{a_1 2} \ldots X^{a_l 2^l} \bmod N$$

Quantum factoring is a good candidate for parallel simulation because it consists of large sections of operations which operate on a subset of the total QU-bits. As shown in Equation 1, the calculation of f(A) consists of a set of multiplications each using only a single bit of A. The simulator can therefore exploit parallelism across the unused bits of A. The simulator can also exploit parallelism within a multiplication because each multiplication is a sequence of additions.

Figure 3 shows in pseudo code the way in which we perform this decomposition for an L bit number. We calculate f(A) as X^{A} bit multiplied by the running product P. This product is calculated for each bit of P into the running sum S using modulo addition. The Modulo addition step comprises the majority of the calculation and operates on a single bit of A and P, and therefore the simulator can exploit parallelism across the unused bits of A and P.

```
For A_bit = A[0].. A[L] /* calculate f(A) as a sequence of
      multiplications */
      P = 1   /* P is the running product */
      For P_bit = P[0] .. P[L-1]       /* calculate a product
          as a sequence of additions */
```

Figure 3 Pseudo Code to calculate f(A)

2.2 Dynamic Allocation of the State Vector

The simulator algorithm uses a hierarchical linked structure to represent the QU-bit vector space. This structure is very similar to a binary tree, except that the number of elements at each level is configurable. The structure uses zero or more link levels and a single level containing the state values. The number of bits which a level covers, determines the size of the block at that level. For the Cirac and Zoller three state model, a block will have 3^B elements for a level covering B QU-bits. The simulator allocates a block for a level when it first uses a bit at that level. This allocation causes the simulator to allocate space for all bits covered by that level. Null pointer values at the link level mark blocks which have not been allocated.

One extreme organization, for the linked structure, is to have a level for every QU-bit. This allocation strategy has a fine granularity and therefore the simulator allocates blocks only when needed. The disadvantage is that the simulator must traverse the maximum number of levels to get to the state values. The other extreme is a flat structure without any link values. This has the disadvantage that the simulator must allocate the complete state space at the start of the simulation. The best compromise is to allocate enough link levels so that bits which are not used until later in the simulation are covered by link levels. This assures that the simulator will allocate the state values for these bits only when the bits are used.

The hierarchical structure has three advantages over a flat structure. The first advantage is that by delaying the allocation of some of the state space, we eliminate unnecessary calculation. This is because all the un-allocated states have an amplitude of zero, and therefore all transformations on them have no effect. The second advantage is that the structure is the same regardless of the number of states used to represent a QU-bit. For example a simulation which uses the detailed three state Cirac and Zoller model has the same structure as a model which uses only two states to represent a QU-bit. The last advantage is that the parallel simulator can reorganize data by redistributing the state value blocks. If the simulator never exploits parallelism across the QU-bits covered by the lowest level, the data in the state value blocks is never split across processors. In a flat structure, to distribute data from one processor to another, the data would not be contiguous and the simulator would have to copy the data to a buffer before sending it. The receiving processor would then have to unpack the data into its' own flat structure.

2.3 Parallel Execution

In a parallel simulation each processor performs a portion of the operations required to implement each of the laser transformations. As described in Section 2.1 the simulator picks a set of bits to parallelize across, and then redistributes the data. Each processor allocates a duplicate copy of all the link blocks in the hierarchical vector structure. A processor however only allocates space for the state values assigned to it.

To perform the operations in a transformation, each processor traverses the linked state vector and performs the calculations on its local states. When all the processors have finished the computation phase, they reorganize the state vector by exchanging the state value blocks. Figure 4 shows the algorithm for redistributing the data between the current organization, defined by the variable current_parallel_bits, and a new organization, defined by the variable new_parallel_bits.

```
For Block_num = 0 .. Num_value_blocks - 1
      current_proc_num = concatenate_parallel_
                bits(Block_num,current_parallel_bits)
      new_proc_num = concatenate_parallel_
                bits(Block_num,new_parallel_bits)
      if ( current_proc_num == new_proc_num ) /* do
      nothing, the data is to remain on same processor */
      else if ( current_proc_num == my_pid ) /* this
            processor currently owns the data */
      send_data_block(new_proc_num) /* send to the
            new owner */
    else /* this processor is to own the data for the next
            step */
    receive_data_block() /* receive the data from the
            current owner */
```

Figure 4. Redistributing the state data

3. APPLICATION ON PRIME FACTORIZATION

Prime Factorization is the core of RSA Public Key Algorithm. If n=pq is factorized, then we can get the $\varphi(n) = (p-1)(q-1)$, and with the Equation 2:

$$SK \times PK \equiv 1 \bmod \varphi(n) \quad \text{(Eq 2)}$$

We can get the secret key SK rapidly.

So the security of RSA algorithm is based on the different of prime factoring. Today the time complexity of the rapidest algorithm is $e^{\sqrt{m(n)\ln\ln(n)}}$.

In 1994, Peter Shor design an algorithm to factorizate large number in polynomial time, convert the NP problem to a P problem. This algorithm converts the prime factorization problem into a new problem to find the period of a function by Quantum Fourier Transform.

With the QCL Paraell Simulator, Intel Pentium Computer and Red Hat Linux OS, we realize the prime factorization algorithm. And a 50bit long number is factorized.

4. CONCLUSION

In this paper we described a parallel simulator on the Intel CPU and RedHat Linux OS which is useful for accessing the feasibility of implementing a quantum computer. Quantum computing is a new field and therefore the simulator is one of the only tools of its kind. And It is an important method to slove some complex hard NP problems such as prime factorization.

5. REFERENCES

[1] R.L. Rivest, A. Shamir, and L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Comm. Assoc. Comput. Mach. 21, p 120. 1978.

[2] P. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. Proceedings, 35th Annual Symposium on Foundations of Computer Science pp.124-134. November 1994.

[3] Kevin M. Obenland and Alvin M. Despain . A Parallel Quantum Computer Simulator. Submitted to High Performance Computing'98, September, 1997

[4] A. Barenco, C. Bennett, R. Cleve et al. Elementary Gates for Quantum Computation Submitted to Physical Review A. March 1995.

On the Convergence of Parallel Chaotic MSOR Method for H-matrix

Dongjin Yuan
Department of Mathematics
Yangzhou University, JiangSu 225002, P. R. of China
E-mail address: dongjinyuan@yahoo.com

ABSTACT

In this paper we establish several algorithms of parallel chaotic MSOR iterative methods for solving large nonsingular systems based on modifying 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.

Key words: Multisplitting, parallel, chaotic, convergence, MSOR method, H-matrix.

1 INTRODUCITON

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

$$A\,x\;=\;b \tag{1}$$

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 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 crutial interest and a number of convergence result (such as [1, 3, 4, 5, 6, 11] etc.) have been obtained. In particular, the convergence of three relaxed chaotic parallel AOR methods have been investigated in references [6, 11]. In this paper, we will establish some Algorithms of chaotic parallel MSOR method and investigate their convergence for $H-$matrices.

Let use consider the linear system (1). We assume that the coefficient matrix A in (1) has Property A (see [16]), that is, A is diagonal or there exists some permutation matrix P such that $P^{-1}AP$ has the form

$$P^{-1}AP=\begin{bmatrix} D_1 & -T \\ -V & D_2 \end{bmatrix}$$

where D_1 and D_2 are nonsingular diagonal matrices, and A is not required to be symmetric. In the sequel, without loss of generality, we assume that

$$A=\begin{bmatrix} I_1 & -T \\ -V & I_2 \end{bmatrix},\quad b=\begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

where I_1 and I_2 are identify matrices of orders i and n-i respectively and partition used for b is in accordance with the partition used for A. Let

$$A = I - L - U \tag{2}$$

with

$$L=\begin{pmatrix} 0 & 0 \\ V & 0 \end{pmatrix} \text{ and } U=\begin{pmatrix} 0 & T \\ 0 & 0 \end{pmatrix}$$

where V and T are respectively the $(n-i)\times i$ *and* $i\times(n-i)$ matrices.

To solve the linear system (1), the MSOR iterative method as in [14], [16] is given by

$$x^{(k+1)} = L_{\omega\omega'}x^{(k)} + g_{\omega\omega'},\quad k = 0,1,2,\cdots \tag{3}$$

where

$$\begin{aligned} L_{\omega\omega'} &= (I-\omega' L)^{-1}[(I-\Omega)+\omega U] \\ &= (I+\omega' L)[(I-\Omega)+\omega U] \end{aligned} \tag{4}$$

and

$$g_{\omega\omega'} = (I-\omega' L)^{-1}d$$

with

$$\Omega=\begin{bmatrix} \omega I_1 & \\ & \omega' I_2 \end{bmatrix}, d=\begin{bmatrix} \omega b_1 \\ \omega' b_2 \end{bmatrix}, \omega\omega' \neq 0. \tag{5}$$

It is clearly that the MSOR iterative method (3) for A is convergent if and only if $\rho(L_{\omega\omega'})<1$.

The main purpose of this paper is to present several Algorithms of relaxed parallel chaotic MSOR 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 NOTATIOM AND ALGORITHUNS

Let us first introduce some of the notation and terminology which will be used in this paper. For

$$x=(x_1,x_2,\ldots,x_n)^T \in R^n \text{ and } A=(a_{ij}) \in R^{n\times n} \text{ by } x\geq 0$$

we mean that $x_i \geq 0$ for $i=1,\ldots,n$,and by $A\geqslant 0$ that $a_{ij}\geq 0$ for $i,j=1,\ldots,n$, in which case we say that x and A are nonnegative. For A, $B\in R^{n\times n}$, we write $A\leqslant B$ 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 $|AB|\leqslant|A||B|$. The notation $\langle A\rangle=(\langle a\rangle_{ij})$ 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.

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 $N \geqslant 0$. For any $s \geq 2$ a multisplitting of $A \in R^{n\times n}$ is a collection of s triple M_l, N_l, E_l of n×n real matrices, $l = 1,2,\cdots,s$, for which each E_l is nonnegative diagonal, each M_l is invertible and the equations

$$A = M_l - N_l, \quad l = 1,2,...,s \tag{6}$$

and

$$\sum_{l=1}^{s} E_l = I \tag{7}$$

are satisfied.

Using the given models in [1,6,11], (3) and (4) we can now describe three Algorithms of relaxed parallel chaotic MSOR method by above notation.

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

$$x^{(k+1)} = \sum_{l=1}^{s} E_l F_l^{\mu_{l,k}}(\omega, \omega', x^{(k)})$$

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

$$with\ \omega > 0,\ \omega' > 0,\ \mu_{l,k} \geq 1.$$

Where $F_l^{\mu_{l,k}}$ is the $\mu_{l,k}$ −th composition of the affine mapping satisfying

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

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,⋯, until convergence, perform

$$x^{(k+1)} = \beta \sum_{l=1}^{s} E_l F_l^{\mu_{l,k}}(\omega, \omega', x^{(k)}) + (1-\beta)x^{(k)}.$$

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

$$\text{with } \beta > 0, \omega > 0, \omega' > 0, \mu_{l,k} \geq 1.$$

Next we will consider the more complicated situation, which is similar to Algorithm 2 in [6] and [11]. In this case the following new terminology is necessary. A sequence of sets P_k with $P_k \subseteq \{1,...,s\}$ is admissible if every integer 1, ⋯, s appears infinitely often in the P_k, while such an admissible sequence is regulated if there exists a positive integer T such that each of the integer 1,⋯, s appears at least once in any T consecutive sets of the sequence. Assume that the index sequence $\{P_k\}$ 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,⋯, until convergence, perform

$$x^{(k+1)} = (I - \beta \sum_{l\in P_i})x^{(k)} + \beta \sum_{l\in P_i} E_l F_l^{\mu_{l,k}}(\omega, \omega', x^{(k-r_i+1)}).$$

$$F_l(\omega, \omega', x^{(k-r_i+1)}) = (I - \omega' L_l)^{-1}[(I - \Omega) + \omega U_l]x^{(k-r_i+1)} + (I - \omega' L)^{-1} d,$$

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

with $\beta > 0, \omega > 0, \omega' > 0, \mu_{l,k} \geq 1, \Phi \neq P_i \subseteq \{1,\cdots,s\}$.

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 [11].

Lemma 3.1 If *A* is an H−matrix, then

(a) $|A^{-1}| \leq \langle A \rangle^{-1}$;

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

$$< A > Pe > 0 \tag{8}$$

with e=(1,⋯,1)^T.

Lemma 3.2 Let A be an M-matrix, and let splitting

A=*M*−*N*

be an M-splitting. If *P* is the diagonal matrix defined in Lemma 3.1, then

$$\| P^{-1}M^{-1}NP \|_\infty < 1 \tag{9}$$

Using above two Lemmas, now we can prove one of our main results, which is a sufficient condition for the convergence of Algorithm 2.1.

Theorem 3.1 Let $A \in R^{n\times n}$ be an H − matrix and $(I - L_l, U_l, E_l), l = 1,2,...,s$, be a multisplitting of A. Assume that for $l = 1,2,...,s$, we have

(1) L_l is the strictly lower triangular matrices and U_l is the matrices such that the equalities A= $I - L_l - U_l$ hold.

(2) $\langle A \rangle = I - |L_l| - |U_l| = I - |B|$,

where $|B| = |L_l| + |U_l|, l = 1,2,...,s$.

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* $(\omega, \omega') \in S_1$,where

$$S_1 = \{(\omega, \omega') \in R^2 : 0 < \omega < 2(1+\rho);$$
$$0 < \omega' < 2/(1 + \rho(|1 - \omega| + \omega\rho))\}$$

with $\rho = \rho(|B|)$.

Proof Let us first define the iterative matrix in the Algorithm 2.1

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

It is clearly that we need to find a constant σ with $0 \leqslant \sigma < 1$ and some norm, which are independent of *k*, such that for

$k \geq 1, \| H(\omega, \omega')_k \| \leq \sigma$.

Since A is an $H-$matrix and for $l = 1,2,...,s, L_l$ is a strictly lower triangular matrix, we see that each $\langle I - \omega' L_l \rangle$ is an $M-$matrix for, $l = 1,2,...s$ and

$$\langle I - \omega' L_l \rangle^{-1} = (I - \omega' | L_l |)^{-1} \geq 0 .$$

Hence each $\langle I - \omega' L_l \rangle$ is an H$-$matrix for $l = 1,2,...,s$, and we have the following inequality

$| (I - \omega' L_l)^{-1} | \leq \langle I - \omega' L_l \rangle^{-1} = (I - \omega' | L_l |)^{-1}$.

From this relation it follows that

$$\begin{aligned} &| (I - \omega' L_l)^{-1} [(I - \Omega) + \omega U] | \\ &\leq \langle I - \omega' L_l \rangle^{-1} | (I - \Omega) + \omega U_l | \\ &\leq (I - \omega' | L_l |)^{-1} [| I - \Omega | + \omega | U_l |]. \end{aligned}$$

Case 1: $0 < \omega \leq 1$ *and* $0 < \omega' \leq 1$. In this case

$$| I - \Omega | + \omega | U_l | = I - \Omega + \omega | U_l | .$$

We denote

$M_l(\omega, \omega') = I - \omega' | L_l |$ and

$$N_l(\omega, \omega') = | I - \Omega | + \omega | U_l | .$$

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

$$\begin{aligned} &M_l(\omega, \omega') - N_l(\omega, \omega') \\ &= I - \omega' | L_l | - I + \Omega - \omega | U_l | \\ &= \Omega - \omega' | L_l | - \omega | U_l | . \end{aligned}$$

Since, for $l = 1,2,...,s, M_l(\omega, \omega')$ are $M-$matrices and $N_l(\omega, \omega') \geq 0$, the splittings $M_l(\omega, \omega') - N_l(\omega, \omega')$ are $M-$splittings of the matrix $\Omega - \omega' | L_l | - \omega | U_l |$, which is clearly the $M-$matrix.

Case 2: $1 < \omega < 2/(1 + \rho)$ and $1 < \omega' < 2/(1 + \rho(| 1 - \omega | + \omega \rho))$. Suppose $\alpha = \max(\omega, \omega')$.

We have

$$| I - \Omega | + \omega | U_l | \leq (\alpha - 1) I + \alpha | U_l | ,$$

and

$$(I - \omega' | L_l |)^{-1} \leq (I - \alpha | L_l |)^{-1} .$$

We also denote

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

and $N_l(\omega, \omega') = (\alpha - 1) I + \alpha | U_t |$,

then $M_l(\omega, \omega') - N_l(\omega, \omega') = (1 - | 1 - \alpha |) I - \alpha | B |$.

It is easy to show that (see from [6] and [11]) $(1 - | 1 - \alpha |) I - \alpha | B |$ is an M-matrix. Since, for $l = 1,2,\cdots,s, M_l(\omega, \omega')$ are M-matrix and $N_l(\omega, \omega') \geq 0$,the splittings $M_l(\omega, \omega') - N_l(\omega, \omega')$ are M-splittings of the matrix $(1 - | 1 - \alpha |) I - \alpha | B |$.

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

$\| P^{-1} M_l^{-1}(\omega\omega') N_l(\omega\omega') P \|_\infty < 1, \quad l = 1,2,...,s$.

and hence

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

Which implies

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

Consequently

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

Let us denote

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

then

$$\| H(\omega, \omega')_k \| \leq \sigma < 1 .$$

We have completed the proof.

Theorem 3.2 Let $A \in R^{n \times n}$ be an H $-$ matrix and $(I - L_l, U_l, E_l), l = 1,2,...,s$,be a multisplitting of A. Assume that for $l = 1,2,...,s$, we have

(1) L_l is the strictly lower triangular matrices and U_l is the matrices such that the equalities $A = I - L_l - U_l$ hold.

(2) $\langle A \rangle = I - | L_l | - | U_l | = I - | B |$,

where $| B | = | L_l | + | U_l |, l = 1,2,...,s$.

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

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 $(\beta, \omega, \omega') \in S_2$, where

$S_2 = \{(\beta, \omega, \omega') \in R^3 : 0 < \beta < 2/(1 + \theta); 0 < \omega < /(1 + \rho);$
$0 < \omega' < 2/(1 + \rho(| 1 - \omega | + \omega \rho))\}$

with

$\rho = \rho(| B |), \theta = \max_{1 \leq l \leq s} \| P^{-1} M_l^{-1}(\omega, \omega') N_l(\omega, \omega') P \|_\infty$

Proof Let us define the iterative matrix in the Algorithm 2.2

$$H(\beta, \omega, \omega')_k = \beta H(\omega, \omega') k + (1 - \beta) I$$

where

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

Similar to the proof of Theorem 3.1 we only need to prove that there exists a constant σ with $0 \leqslant \sigma < 1$, which is independent of k, such that

$$\| P^{-1} H(\beta, \omega, \omega')_k P \|_\infty \leq \sigma .$$

From the relation in the proof of Theorem 3.1

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

we obtain

$$\begin{aligned} &\| P^{-1} H(\beta, \omega, \omega')_k P \|_\infty \leq \beta \| P^{-1} H(\omega, \omega')_k P \|_\infty + | 1 - \beta | \\ &\leq \beta \max_{1 \leq l \leq s} \| P^{-1} M_l^{-1}(\omega, \omega') N_l(\omega, \omega') P \|_\infty + | 1 - \beta |. \end{aligned}$$

Clearly, if ω, ω' and β satisfy the condition of this Theorem then

$\alpha \equiv \beta \max_{1\le l\le s} \| P^{-1}M_l^{-1}(\omega,\omega')N_l(\omega,\omega')P\|_\infty + |1-\beta| < 1$

which completes the proof.

Using the proving process of Theorem 3.1, 3.2 and [11, Theorem 2.8] we get the following convergence of Algorithm 2.3.

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

(1) L_l is the strictly lower triangular matrices and U_l is the matrices such that the equalities $A = I - L_l - U_l$ hold.

(2) $\langle A\rangle = I - |L_l| - |U_l| = I - |B|$,

where

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

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

(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 $(\beta,\omega,\omega') \in S_3$, where

$$S_3 = \{(\beta,\omega,\omega') \in R^3 : 0 < \beta < 2/(1+\theta);$$
$$0 < \omega < 2/(1+\rho);$$
$$0 < \omega' < 2/(1+\rho(|1-\omega|+\omega\rho))\}$$

with

$$\rho = \rho(|B|), \theta = \max_{1\le l\le s} \| P^{-1}M_l^{-1}(\omega,\omega')N_l(\omega,\omega')P\|_\infty$$

4 REFERENCES

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

[5] A. Frommer and G. Mayer, *Convergence of relaxed parllel multisplittings methods*, Linear AlgebraAppl. , 199 (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. Neuman 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 Dicrete Methods. , 6(1985), 630－640.

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

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

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

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

[14] X. Wang, *Convergence theory for the general GAOR type iterative method and the MSOR iterative method applied to H－matrices*, Applied Numerical Mathematics, 21 (1996) 469－479.

[15] X. Wang, *Convergence theory for the general GAOR type iterative method and the MSOR iterative method applied to H－matrices*, Linear Algebra Appl. , 250(1997), 1－19.

[16] D.M. Yong, *Iterative Solution of Large Linear Systems*, Academic press, 1971.

A Novel Incremental Algorithm for Mining Frequent Itemsets

Yunlan Wang, Zengzhi Li, Jun Xue, Shimin Ban, Yinliang Zhao
Institute of Computer Architecture and Networks, Xian Jiaotong University, Xi'an, China, 710049
E-mail: wangyunlan@263.net

ABSTRACT:

A novel efficient incremental algorithm IAA is presented for maintaining the frequent itemsets discovered in a database in the cases including insertion, deletion and modification of transactions. It utilizes a heuristic selective scan technique to reduce the number of database scans required and to keep the size of the candidate itemsets sets from increasing quickly. The useful relations between the previous frequent itemsets with respect to the origin database and the new frequent itemsets with respect to the updated database are studied, that is used to prune the candidate itemsets efficiently when scanning the added and deleted portions of the database. The proposed algorithm has been implemented and its performance is studied. It is shown that the option of selective scan and prune technique is very advantageous and can lead to prominent performance improvement.

Keywords: KDD, Data Mining, Frequent Itemsets, Association Rules, Incremental mining

1. INTRODUCTION

The importance of data mining is growing at rapid pace recently. Analysis of the past transactions data can provide very valuable information on customer behavior and business decisions. Of the various data mining problems, mining of association rules is an important one. The following two problems are essential in order to make the database mining a feasible technology. Firstly, design efficient algorithms for mining rules or patterns from database system; secondly, design efficient algorithms to update, maintain and manage the rules discovered.

The first problem has been studied substantially with many interesting and efficient data mining algorithms reported. Apriori [1] runs a number of iterations and in each iteration scan the database once to compute the frequent itemsets of the same size. DHP [2] is the improvement of Apriori in which the hash technique is adopted. TreeProjection [3] is an algorithm for generation of frequent itemsets by successive construction of the nodes of a lexicographic tree of itemsets. FP-tree [4] proposes a novel frequent pattern tree structure and develops an efficient FP-tree based mining method, which only need to scan the database once and is efficient for small databases. All of the algorithms above have not considered the problem of the refresh of the database.

FUP [5] is an incremental updating technique for efficient maintenance of discovered association rules when new transactions are added to the transaction database. FUP2 [6] is a complementary algorithm of FUP, which is efficient when the deleted portion is a small part of the database. FUP and FUP2 tend to suffer from the inherent problem of need multiple scans of database. SWF [7] is a sliding-window filtering algorithm for incremental mining, which partition the transactions database into several partitions and only need to scan the updated database once. But it will incur the huge size of candidate itemsets.

In this paper, we presented a new incremental algorithm, called IAA. The trait of IAA is as follows: 1) The selective scan technique is adopted to determine frequent itemsets in batch so as to reduce the number of database scans required and to avoid the size of the candidate set becomes huge too quickly. 2) IAA is a general incremental algorithm for maintaining the frequent itemsets in the cases of insertion, deletion and modification of the transactions in the database. It takes advantage of the previous mining result to cut down the cost of finding the new rules in a frequently updated database. 3) The useful relations between the previous frequent itemsets with respect to the origin database and the new frequent itemsets with respect to the updated database are studied, that is used to prune the candidate itemsets efficiently when scanning the added and deleted portions of the database.

The remainder of the paper is organized as follows: The new incremental mining algorithm for frequent itemsets is presented in section 2. The performance study of the incremental algorithm IAA is reported in section 3. The conclusions are presented in section 4.

2. THE INCREMENTAL DATA MINING ALGORITHM FOR FREQUENT ITEMSETS

2.1 Problem Statement

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of items, let D be a database of transactions, where each transaction is a set of items such that $T \subseteq I$. Given an itemset $X \subseteq I$, a transactions T contain X if and only if $X \subseteq T$. For an itemset X, its support is defined as the percentage of the transactions in D that contain X. Its support count, denoted as $X_{.\sup}$, is the number of transactions in D that containing X. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subseteq I$, $Y \subseteq I$ and $X \bigcap Y = \phi$. The association rule $X \Rightarrow Y$ holds in D with confidence c% if c% of the transactions in D that contain X also contains Y. The association rule $X \Rightarrow Y$ has support s% if s% of the transactions in D that contain $X \bigcup Y$. Given a minimum confidence threshold and a minimum support threshold the problem of mining association rules is to find all the association rules that have confidence and support greater than the corresponding threshold. This problem can be reduced to the problem of finding all frequent itemsets for the same support threshold [1].

The large amount of transactions in the database and the mining results poses a maintenance problem. While new transactions are being appended and obsolete ones are being removed, association rules already discovered also have to be updated. This paper in devoted to the research of incremental algorithm that reduces the size of candidate set and the number of database scans required.

2.2 The Selective Scan Technique for Mining Frequent

Supported by the National Natural Science Foundation of China under Grant No.60173066.

Itemsets

It is observed that in some applications the length of a transaction could be very large. In this case, the Apriori-like algorithms, which scan the database iteratively, such as DHP, FUP and FUP2, will have to scan the database many times, which degrades the execution efficiency. On the other hand, the algorithm SWF adopted the scan reduction technique that generates $C_2^{'}$ from the scan of the added and deleted portion of the database and generates $C_k^{'}(k>2)$ from $C_{k-1}^{'} * C_{k-1}^{'}$ instead of from $L_{k-1}^{'} * L_{k-1}^{'}$, where $C_k^{'}(L_k^{'})$ is the candidate (frequent) *k*-itemsets in the updated database $D^{'}$. It finds the $L_k^{'}$ $(k=1,2,...,n)$ together when the scan of the updated database $D^{'}$. Note that when the minimum support is relatively small, the size of $C_k^{'}$ could become huge quickly. It may cost too much CPU time to generate all candidate itemsets and all frequent itemsets. To overcome the shortcoming of FUP2 and SWF, we adopt a heuristically selective scan strategy, the idea is similar to the algorithm SS in reference [8] but the condition to scan the unchanged portion of the database is more sound and efficient than that is used in SS. The selective scan strategy used in IAA is: if $k=1$ or $|Q_k|*k+|C_{k+1}|*k*2>|Q_{k-1}|*(k-1)$ (where $Q_k=C_k^{'}-C_k^{'}\bigcap L_k$), we scan the database once to obtain the frequent itemsets that have not generated in the current and previous iterations. Otherwise, we do not scan the unchanged portion of the database and generate $C_{k+1}^{'}$ from $(P_k\bigcup Q_k)*(P_k\bigcup Q_k)$. The selective scan technique can reduce the number of database scan required efficiently and at the same time the size of candidate itemsets will not increased quickly.

2.3 The Useful Lemma and Corollary for Incremental Mining

After some update activities, old transactions are deleted from the database and new transactions are added. We can treat the modification of existing transactions as deletion followed by insertion. Let *D* be the database before update, d^+ be set of newly added transactions and d^- be the set of deleted transactions. Let $D^{'}$ be the updated database, $D^{'}=(D-d^-)\bigcup d^+$. Let D^- be the unchanged portion of the database, $D^-=D-d^-$. Let *s* be the minimum support threshold. We use $X_{.\sup}$, $X_{.\sup-D}$, $X_{.\sup+}$, $X_{.\sup-}$, $X_{.\sup-D^-}$ to denote the support count in $D^{'}$, D, d^+, d^-, D^- respectively. And we use $|D|$, $|d^+|$, $|d^-|$, $|D^{'}|$ to denote the number of transactions in *D*, d^+, d^- and $D^{'}$ respectively. Let *L* be the set of itemsets that are previously frequent with respect to *D* and $L^{'}$ be the set of frequent itemsets with respect to the updated database $D^{'}$.

Inspired by the reference [6], we study and use the following lemmas and corollaries in the process of mining the frequent itemsets with respect to the updated database $D^{'}$.

Lemma 1: An itemset X is frequent in the updated database $D^{'}$ if and only if: $X_{.\sup}\geq s*|D^{'}|$.

Where $X_{.\sup}=X_{.\sup-D}+X_{.\sup+}-X_{.\sup-}$

$$|D^{'}|=|D|+|d^+|-|d^-|$$

Prof: by definition of frequent itemset.

Corollary 1: An itemset X is not frequent in the updated database $D^{'}$ if:

$$X_{.\sup-D}+X_{.\sup+}<s*|D^{'}|$$

Lemma 2: If $X\notin L$ and $X\in L^{'}$, it must satisfy:

$$X_{.\sup+}-X_{.\sup-}>s*(|d^+|-|d^-|).$$

Prof: Since $X\notin L$ and $X\in L^{'}$,

We have:

$X_{\sup-D}<s*|D|$ and

$$X_{.\sup-D}+X_{.\sup+}-X_{.\sup-}\geq s*|D^{'}|$$

Hence,

$$\begin{aligned}&X_{.\sup+}-X_{.\sup-}\\&\geq s*|D^{'}|-X_{.\sup-D}\\&>s*(|D|+|d^+|-|d^-|)-s*|D|\\&=s*(|d^+|-|d^-|)\end{aligned}$$

Thus we have the lemma.

Corollary 2: If $X\notin L$ and $X_{\sup+}\leq s*(|d^+|-|d^-|)$, then $X\notin L^{'}$.

2.4 The Novel Incremental Algorithm IAA

To discover the frequent itemsets in the updated database, the algorithm IAA executes iteratively. In the *k*-iteration, the candidate set $C_k^{'}$ is divided into two parts: $P_k=C_k^{'}\bigcap L_k$ and $Q_k=C_k^{'}-P_k$. In other words, $P_k(Q_k)$ is the set of candidate itemsets that are previously frequent (not frequent) with respect to D. Our goal is to select those itemsets that are frequent in the updated database $D^{'}$. We treat the candidates in these partitions separately. With this partitioning, for all candidates $X\in P_k$, we already know it support count $X_{.\sup-D}$ from the previous mining results. $X_{.\sup+}$ can be got by scanning d^+, so the itemsets in the set

$$\{X|X_{.\sup-D}+X_{.\sup+}<s*|D^{'}|,X\in P_k\}$$

are deleted from P_k because they cannot be frequent in $D^{'}$ according to corollary 1. We can find out $X_{.\sup-}$ by scanning d^-, so the itemsets in the set

$$\{X|X_{.\sup-D}+X_{.\sup+}-X_{.\sup-}<s*|D^{'}|,X\in P_k\}$$

are deleted from P_k according to lemma 1.

For the itemsets $X\in Q_k$, we know that they are not frequent in the original database *D*. We can find out the $X_{.\sup+}$ by scanning d^+, so the itemsets in the set

$$\{X|X_{.\sup+}<s*(|d^+|-|d^-|),X\in Q_k\}$$

are deleted from Q_k because they cannot be frequent in the updated database $D^{'}$ according to corollary 2; Then we can find out the $X_{.\sup-}$ by scanning d^-, so the itemsets in the set

$$\{X|X_{.\sup+}-X_{.\sup-}<s*(|d^+|-|d^-|),X\in Q_k\}$$

are deleted from Q_k because they cannot be frequent in the updated database $D^{'}$ according to lemma 2.

The selective scan and candidate generate method is as

follows: If $k=1$ or $|Q_k|*k+|C_{k+1}|*k*2>|Q_{k-1}|*(k-1)$, we scan the unchanged portion D^- to get the set of frequent itemsets $\bigcup_{i=k_0}^{k} L_k$ in batch. k_0 and k denote that we have not scan D^- from k_0 to k iteration. The candidate set C_1' is the set of all the items. C_k' is generated from $P_{k-1}\bigcup Q_{k-1}$ or L_{k-1}'. If k is equal to k_0, C_k' is generated from L_{k-1}'. Otherwise, C_k' is generated from $P_{k-1}\bigcup Q_{k-1}$.

In the following, we illustrate the incremental algorithm IAA in detail. Before that, for clarity purpose, we list the notations used in our discussion in Table 1.

Input: D, d^+ and d^-, L, $X_{.\sup-D}$ for $X\in L$. Support threshold s.

Output: L', $X_{.\sup}$ for $X\in L'$.

Algorithm IAA

1 C_1'= I; $k=1$; $k_0=1$; //I is the set of items.
2 while $C_k'\neq\phi$
3 $P_k=C_k'\bigcap L_k, Q_k=C_k'-P_k$
4 Scan d^+ to get $X_{.\sup+}$ for each $X\in P_k\bigcup Q_k$
5 for each itemset $X\in P_k$
6 if $X_{.\sup-D}+X_{.\sup+}<s*|D'|$
7 $P_k=P_k-\{X\}$;// corollary 1
8 end if
9 end for
10 if $|d^+|>|d^-|$, for each itemset $X\in Q_k$
11 if $X_{.\sup+}\le s*(|d^+|-|d^-|)$
12 $Q_k=Q_k-\{X\}$;// corollary 2
13 end if
14 end if, end for
15 Scan d^- to get $X_{.\sup-}$ for each $X\in P_k\bigcup Q_k$
16 for each itemset $X\in P_k$
17 if $X_{.\sup-D}+X_{.\sup+}-X_{.\sup-}<s*|D'|$
18 $P_k=P_k-\{X\}$; // Lemma 1
19 end if
20 end for
21 for each itemsets $X\in Q_k$
22 if $X_{.\sup+}-X_{.\sup-}\le s*(|d^+|-|d^-|)$
23 $Q_k=Q_k-\{X\}$; // Lemma 2
24 end if
25 end for
26 if $k=1$ or $|Q_k|*k+|C_{k+1}|*k*2>|Q_{k-1}|*(k-1)$
27 Scan D^- to get $X_{.\sup\_D^-}$ for each $X\in\bigcup_{i=k_0}^{k}Q_i$
28 $L_i'=P_i\bigcup\{X|X_{.sup}\ge s*|D'|, X\in Q_i\}$, $k_0\le i\le k$
29 $k_0=k+1$;
30 end if
31 $k++$;
32 if $k==k_0$
33 /* Produce C_k' from L_{k-1}' */
34 $C_k'=apriori\_gen(L_{k-1}', s)$
35 else
36 /* Produce C_k' from $P_{k-1}\bigcup Q_{k-1}$ */
37 $C_k'=apriori\_gen(P_{k-1}\bigcup Q_{k-1}, s)$
38 end if
39 end while
40 if $k>k_0$
41 Scan D^- to get $X_{.\sup\_D^-}$ for $X\in\bigcup_{i=k_0}^{k-1}Q_i$
42 $L_i'=P_i\bigcup\{X|X_{.sup}\ge s*|D'|, X\in Q_i\}$, $k_0\le i\le k$ -1
43 end if

Table 1 Notation Table.

D	The origin database
d^+	The added portion to the database
d^-	The obsolete portion of the database
D'	The updated database $D'=(D-d^-)\bigcup d^+$
D^-	The unchanged portion of the database $D^-=D-d^-$
$\|D\|$	The number of transactions in D
$\|d^+\|$	The number of transactions in d^+
$\|d^-\|$	The number of transactions in d^-
$\|D'\|$	The number of transactions in D' $\|D'\|=\|D\|+\|d^+\|-\|d^-\|$
$X_{.\sup}$	The support count of an itemset X in D'
$X_{.\sup-D}$	The support count of an itemset X in D
$X_{.\sup+}$	The support count of an itemset X in d^+
$X_{.\sup-}$	The support count of an itemset X in d^-
$X_{.\sup-D^-}$	The support count of an itemset X in D^-
L	The set of frequent itemsets in D
L_k	The set of frequent k-itemsets in D
L'	The set of frequent itemsets in D'
L_k'	The set of frequent k-itemsets in D'
C_k'	The set of candidate k-itemsets in D'
P_k	$C_k'\bigcap L_k$
Q_k	$C_k'-P_k$

3. PERFORMANCE STUDY OF IAA

To assess the performance of the algorithm IAA, we performed several experiments on a computer of 2.0GHz and 512 MB of memory. The simulation program was coded in C++. The first experiment is to compare the time performance of IAA to that of Apriori and FUP2. The next experiment is to find out how the size of d^+ and d^- affects the performances of the algorithms. We use the notation Tt.Ii.Dm-x+y to

denote an updated database in which $|D| = m$, $|\mathrm{T}| = t$, $|\mathrm{I}| = i$, $|d^-| = x$ and $|d^+| = y$ where $|D|$ is the number of transactions in the origin database, $|\mathrm{T}|$ is the mean size of the transactions and $|\mathrm{I}|$ is the mean size of the potentially frequent itemsets, x is the number of transactions added to the database and y is the number of transactions deleted from the database.

3.1 Relative performance

In the first experiment, the execution time is compared between Apriori, FUP2 and IAA with respect to T5.I4.D10k-1k+2k. The support threshold is varied between 0.5% and 4.0%. The results are plotted in figure 1. It can be seen that IAA is faster than the FUP2 algorithm 8% to 34% and is faster than Apriori 0.74 to 2.13 times.

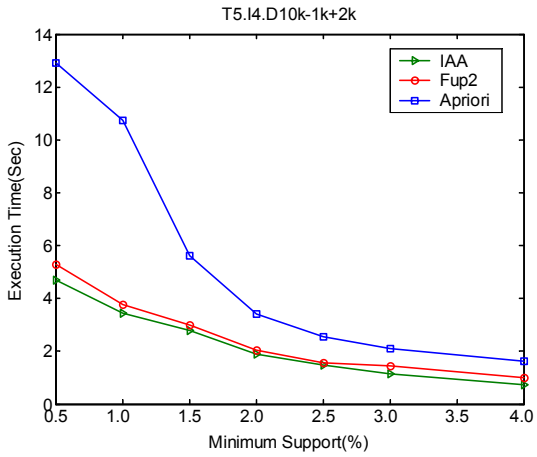


Figure 1 Relative Performance

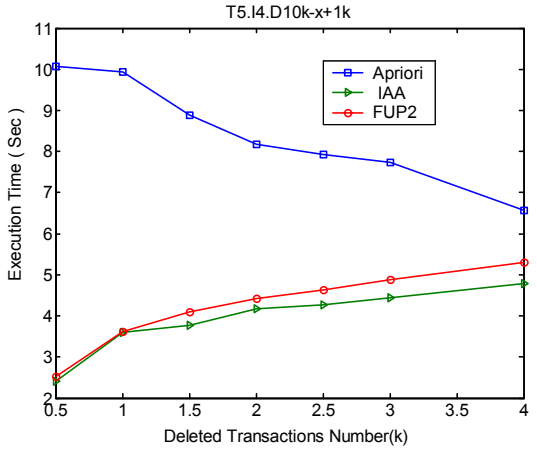


Figure 2 Effect of the Size of Deleted Portion

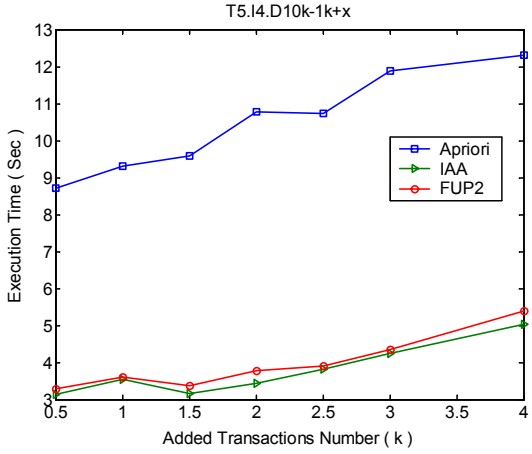


Figure 3 Effect of the Size of Added Portion3.2 Effect of the Size of Deleted Portion and the Size of Added portion

Our next experiment is to find out the impact of $|d^-|$ to the performance of algorithms. We use the setting T5.I4.D10k-x+1k for the experiment, with a support threshold of 1.0%. In other words, we use an initial database of 10 thousand transactions. From this database, x thousand transactions are deleted, x is ranged form 0.5 to 4.0, and another one thousand transactions is added to the database. Figures 2 shows the results of this experiment. As the number of deleted transactions increases, the amounts of time taken by Apriori decrease, since the size of updated database decreases. As x increase, the number of transactions that IAA and FUP2 have to handle increase; therefore, these algorithms spend more and more time as x grows. However, IAA and FUP2 still outperform Apriori for $x < 4.0$. Furthermore, IAA is faster than FUP2 2% to 11% and is faster than Apriori 0.37 to 3.18 times.

A similar experiment is done using the setting T5.I4.D10k-1k+x and the same support threshold of 1.0%. This time, we keep $|d^-|$ constant and vary $|d^+|$. The results are shown in figure 3. As x increase, $|D'|$ increases. So the execution time of Apriori increases with x. IAA and FUP2 also spend more and more time when x grows. It can be seen that IAA is faster than the FUP2 algorithm 3% to 10% and is faster than Apriori 1.45 to 2.14 times.

4. CONCLUSIONS

In this paper, we have proposed a novel efficient incremental algorithm for mining frequent itemsets. The selective scan techniques is utilized to reduce the number of database scan required and to keep the set of candidate itemsets from becoming huge quickly. It studies the useful property in incremental mining to found some lemmas and corollaries that are adopted to prune the candidate set efficiently. The performance of the algorithms is studied. The results show that the algorithm is validity and has superior performance.

5. REFERENCES

[1] R. Agrawal, T. Imielinski, A. Swami. “Mining Association Rules between Sets of Items in Large Databases”. Proceeding of ACM SIGMOD International Conference on Management of Data, 1993, pp.207-216.

[2] J.S. Park, M. S. Chen, and P. S. Yu. “Using a Hash-Based Method with Transaction Trimming For Mining association Rules”. IEEE Transactions on Knowledge and Data Engineering, vol. 9, 1997, pp. 813-825.
[3] R. Agarwal, C. Aggarwal et al. “A Tree Projection Algorithm for Generation of Frequent Itemsets”. Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining), March 2001, pp.1-23.
[4] Jiawei Han, Jian Pei, and Yiwen Yin, “Mining Frequent Patterns without Candidate Generation”. Proceeding of ACM SIGMOD International Conference on Management of Data, 2000, pp.1-12.
[5] D. W. Cheung, J. Han, V. T. Ng and C.Y. Wong. "Maintenance of Discovered Association Rules in Large Database: An Incremental Updating Technique". Proceeding of Twelfth Conference On Data Engineering (New Orleans, Louisiana), 1996, pp.106-114.
[6] D. W. Cheung, S.D. Lee, Benjamin Kao. "A General Incremental Technique for Maintaining Discovered Association Rules". Proceedings of the Fifth International Conference on Database Systems for Advanced Applications, 1997, pp. 185-194.
[7] Chang-Hung Lee, Cheng-Ru Lin, and Ming-Syan Chen. “Sliding Window Filtering: An Efficient algorithm for incremental mining”. Proceedings of the ACM tenth International Conference on Information and Knowledge Management (Atlanta, Georgia, USA), 2001, pp.263-270.
[8] Ming-Syan Chen, Jong Soo Park and Philip S. Yu, "Efficient Data Mining for Path Traversal Patterns", IEEE Transactions on Knowledge and Data Engineering, Vol.10, No.2, 1998, pp. 209-221.

An Improved Genetic Algorithm For Optimizing Neural Network Weights

Liu Lan
College of information Engineering, Wuhan University of Technology
Wuhan, HuBei, China
E-mail: whekon@public.wh.hb.cn
And
Wu wei
College of information Engineering, Wuhan University of Technology
Wuhan, HuBei, China
E-mail: freebird2200@163.com

ABSTRACT

This article introduces a kind of improved genetic algorithms (IGA) to optimize neural network weights. The instance indicated that the method of applying genetic algorithms to network could improve the performance of network in effect.

Key words genetic algorithm, weight, fitness function, neural network, local optimum

1. INTRODUCTION

NN (Neural network) is widely used in many fields such as intelligent control, system optimization, signal and information processing, pattern recognition, etc, in which multilayer NN is one of the most important NN models that widely used in the fields of signal processing and pattern recognition. BP algorithm is the most typical learning algorithm [1] for multilayer NN and succeeds in local search. Because BP algorithm mostly adopts error derivative to guide learning process, being local optimizing algorithms, and most BP networks use the searching algorithms of gradient descent, the following problems are inevitable [2]: (1) when the networks learn with BP algorithms, learning outcome is abnormally sensitive to initial weight vector, different initial weight vector may result in quite another outcome; (2) If there are more local minimum when modeling for complex problem, the system is easy to be trapped into local minimum point; (3)During the computing process, some parameters such as training speed and inertia coefficient could only be lied on experiment and experience, choosing the unsuitable parameters might bring oscillation and can't be convergent. Furthermore, this algorithm even has some shortcomings such as slow convergence speed, non ideal dynamic characteristic, inconsistency in learning precision and speed, etc.

GA (Genetic Algorithms) is a kind of self-adapting searching and machine learning process forming by simulating the law of evolution that "survival of the fittest in natural selection" of biome in nature. It's mainly applied into function optimizing [3]. Since GA must keep out certain scale colony, instead of processing single point like gradient descent algorithm, it has to search umpty points in space synchronously to hunt global optimum point and avoid the trap of local minimum. In this way, the shortcomings of gradient descent algorithm could be avoid efficiently. On the other hand, different from blind searching and entire random searching, GA adopts crossover operator. It's a process of constructing better solution based on current optimum, and therefore it has the virtues of certain heuristic search and high searching efficiency. Furthermore, the massive parallelism of GA attracts people to apply the GA to NN and get some achievement.

The weight training of NN is actually a optimization problem, namely, searching optimum continuous weight and minimizing the difference between the outcome of NN and destination function. So GA is good to training the neural network weights. Traditional GA is easily trapped into local optimum for its faint ability of climbing, an improved genetic algorithm is introduced to optimize neural network weights in this paper.

2. THE FUNDAMENTALS OF TRAINING BP NEURAL NETWORK WEIGHTS BY IGA

GA optimize the weight training of BP network as follows [4]:

(1) Generate a set of initial weights at random, and code them into individuals of genotype in genetic space;

(2) Compute error function based on the weights of NN produced, and get the fitness function.

(3) Make some genetic operations such as selection, crossover, variation, etc. On current colony, generate the next generation colony, and the individual with maximum value of fitness function is entailed on next generation directly.

(4) If a set of weights could be satisfied with the accuracy specification, then finish, else turn to step (2).

Let's take the multi-layer NN model showed in figure 1 for example. Assume the number of training sample is *P*, current learning sample is *p*, then the net input of node j is defined as:

$$I_{pj} = \sum_j W_{ij} \bullet O_{pj} - \theta_j \tag{1}$$

Where O_{pj} denotes the output of node *j* of previous layer, W_{ij} network connection weight, and θ_j threshold value of node *j*. The output of node j can be express as:

$$O_{pj} = f(I_{pj}) \tag{2}$$

where $f(\bullet)$ is a transform function. If choosing sigmoid function, that is $f(x) = 1/(1+e^{-x})$, then Eq. (2) transform to the following expression:

$$O_{pj} = \frac{1}{1+e^{-I_{pj}}} \tag{3}$$

If the error function of training network weight is defined as :

$$E_p = \sum_j (t_{pj} - O_{pj})^2 \tag{4}$$

where t_{pj} is ideal output of node *j*, then error function can be calculated by Eq. (4)

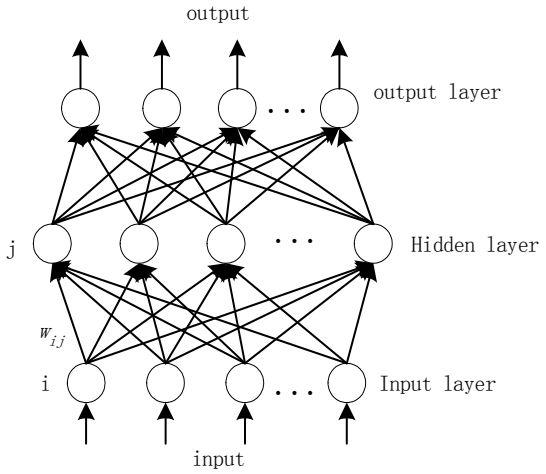


Fig. 1 Multi-layer forward NN model

Now, we ascertain the value of fitness function. The choice of fitness function can guide search space gradually approaching optimum parameter combination along the direction of optimizing parameter combination, avoiding divergence and the trap of local optimum. In the searching process, fitness function is used to evaluate the good and bad of each chromosome, and the chromosome with higher function value represents better solution. Choose the chromosome with higher fitness function value to proceed regeneration, generate new generation of genome which could be more adapted to the environment through operations of crossover and variation. In this way, it evolves generation after generation until it converges to an individual adapted to the environment mostly, and gets the optimal solution.

Optimization problem is a problem of minimal value. Assumed that $f(x)$ is the fitness function of certain individual in current generation, $g(x)$ is the destination function of certain individual in current generation, $g_{\max}$ is the maximum destination function in current generation, then the fitness function is defined as:

$$f(x) = g_{\max} - g(x) \tag{5}$$

In the searching process, the chromosome with higher fitness function value represents better solution. If the fitness function value of an individual in current generation is quite high, it will be easy to be trapped into local optimum. To prevent searching process from getting into local optimum, the fitness function in this paper is modified as follows:

$$f'(x) = f(x) - F + K \cdot g_{\min} \tag{6}$$

Where $f_i'(x)$ denotes the modified fitness function of a certain individual in current generation, $f(x)$ is the fitness function of this individual in current generation, F is the average value of destination function value in current generation, $g_{\min}$ is the minimal destination function value in current generation, The experience parameter K is usually taken 1~3. In this way, the certain fitness function in a certain generation can be prevented from being too high and it could be driven out of the trap of local optimum.

The process of training NN with IGA is illustrated in Fig.2, in which E is the given error precision.

3. THE INSTANCE FOR TRAINING NEURAL NETWORK WITH IGA

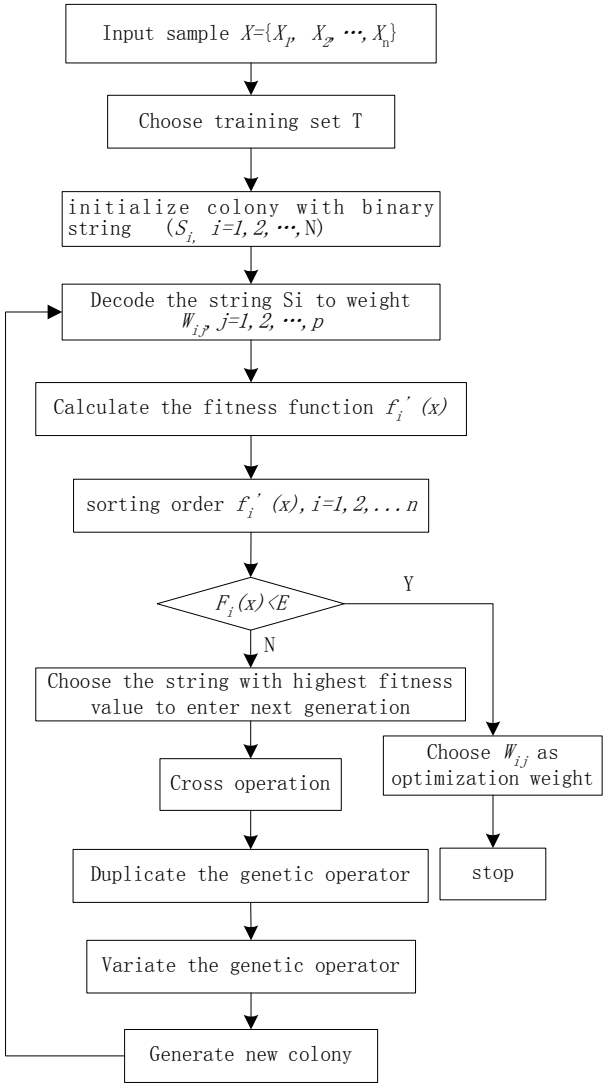


Fig. 2 The process of training the weight of NN with IGA

XOR is always a problem which could not be solved perfectly by traditional BP algorithm [5]. Here we use IGA to evolve the connection weight of the problem of XOR.

XOR problem is a noted instance in neural network, which is a typical problem of linear impartibility. Hidden layers are needed in NN, and here we use a multilayer perception with one hidden layer to learn XOR problem. Fix the structure of NN is $N_{2,2,1}$, the input sample is {{0,0},{0,1},{1,0},{1,1}}, the corresponding output is {{0},{1},{1},{0}}, the excitation function of the neuron of hidden layer and output layer is a asymmetric function defined as $f(x) = 1/(1+e^{-x})$.

The implementation steps of training the weight of XOR network with IGA are given as follow:

(1) Use the project of binary coding. In the network, there are six weights that are W_{13}, W_{14}, W_{23}, W_{24}, W_{35}, W_{45} and three threshold values. Assume that each weight (or threshold value) is represent by a 0/1 string of 10 bit, and a 0/1 string of 90 bit is correspond to a weight of NN, including distribution of threshold value, For simplicity, the threshold value could be given as zero.)

(2) Compute the error function to ascertain the fitness function, the more is the error, the smaller is the fitness value. In this case, the fitness function is defined as $f'(x) = f(x) - F + K \cdot g_{\min}$

(3) Choose the chromosome. Take the selective probability as: $P_s = \frac{f_i}{\sum_{j=1}^{N} f_j}$, the individual with maximum value of fitness function is entailed next generation directly.

(4) Cross the chromosome. Use the cross probability P_c, and proceed the cross calculation with one-point-cross.

(5) Invoke duplicate operator to reorganization the filial generation after crossover.

(6) Invoke variation operator to proceed variation operation on the *i*th chromosome in filial generation.

(7) Repeat steps (2)~(6), make the initial group of weight distribution evolve continuously, until it achieves the number of predeterminate evolution generation or the error *e* is less than some given value E.

Here we assume that the crossover probability P_c=0.05, the variation probability P_m=0.03, the maximal generation number N=50. Set the parameter of BP algorithm as: rate of learning η =0.7, coasting ability a =0.9. The experimental results are listed in table 1, in which the amount of calculation is counted following the method introduced in ref [6].

Table 1 The network training degree of three kind of algorithms for XOR problem

mean square error	BP algorithm	Combination of traditional GA and BP algorithm	Combination of IGA and BP algorithm
0．1	1140	25	16
0．01	1350	32	23
0．001	3740	323	314

From the training result, training XOR problem with IGA can get higher convergence speed and less time for achieving required accuracy. Furthermore, the application of IGA to training NN weight can avoid the trap of local minimum in effect. As a result, IGA with modified fitness function could get better application effect for optimizing neural network weights in a way.

4. CONCLUSION

From the discussion above, genetic algorithm has powerful processing capacity and optimizing capacity, improved GA can avoid the trap of local minimum more efficiently and quicken the convergence speed of weight training. Applying IGA to optimizing the weight of BP NN can get positive sense for the generalization and utility of NN.

5. REFERENCES

[1] Liu Shuguang, Zheng Congxun, Liu Mingyuan. BP algorithm in feedforward neural network and its modification, Computer Science, Vol.23, No.1, May 1996, pp.76-79.

[2] Shu Yunxing, Zhang Yongsheng, Yu Ke. Study on Back propagation Network optimization with genetic algorithms, Journal of ShanDong institute of building materials, Vol.14, No.1, March 2000, pp.22-24.

[3] Holland JH.Outline For Logical Theory Of Adaptive Systems, Journal of Association For Computer Machinery, Vol.25, No.3, March 1962, pp.297-314.

[4] Ge ling，Wu Xinyu.BP Neural Network Optimization using Adaptive Genetic Algorithm, journal of Nanjing Institute of posts and Telecommunication, Vol.18, No.3, July 1998, pp.1-4.

[5] Chen Guoliang. Genetic algorithms and its application, Beijing: pptph.,1996.

[6] Melanie. Mitchell. An introduction to Genetic Algorithms, Cambridge Massachusetts: MIT press. 1992.

Association Rule Algorithm Based on Frontier Set

Zhang Yufang, Xiong Zhongyang, Hu Yue
Department of Computer Science, Chongqing University
Chongqing, 400044, China
E-mail: zhangyf@cqu.edu.cn

ABSTRACT

Mining the association rules is an important aspect of the study of data mining. This paper analyzes some problems existing in the mining algorithms of association rules, presumes upon mining on the high level to have more significance meaning. The concept of frontier set and detailed description of revised association rule algorithm are given. Frontier set is the precondition of candidate set. It improves the efficiency through handpicking candidate set. In addition to realizing in sales data of company, some valuable association rules were gained. The memory management problem encountered in the application is also analyzed.

Keywords: Data Mining, Association Rule, Candidate Set, Large Itemset, Frontier Set

1. WHAT IS ASSOCIATION RULE MINING

R. Agrawal proposed the association rule mining of transactional database. It becomes a very important aspect of the study of data mining. Agrawal brought forward association rule problem with customer transaction database in 1994. The main idea of Apriori is level-wise based on frequency theory. A typical example of association rule is 90% customers buy bread together with milk. The association rule can be used for applications range from customer purchase analysis, catalog design, advertising mail strategies, to store layouts and network failure analysis. Association rule mining belongs to classification based on discovering schema, also to rule induce mining technology. Recently, the database of commercial, government and science increased rapidly, so the huge amounts of data are accumulated. All these laid the necessary foundation for data mining. The data mining technology provides reliable scientific method for us to uncovering efficiency, novelty, interesting data patterns hidden in large data sets. It is becoming one of the hot topics in artificial intelligence and database area.

2. THE SUMMARIZE OF ASSOCIATION RULE MINING TECHNIQUE

The task of association rule mining is to discover association rules that satisfy the conditions in a given database. Association rules are of the form $A_1 \Rightarrow A_2$, support = s%, confidence = c% where s is a given support threshold and c confidence threshold. Association rule mining can be carried out in different abstract concept layer. For example, compare R_1: $diaper \Rightarrow beer$, support= 5%, confidence = 50% with R_2: $baby - commodity \Rightarrow soft - drink$, support = 25%, confidence = 80%, obviously R_2 is in higher abstract level, has bigger support and confidence count, more significance, more suitable for decision making.

2.1 The Formal Description of Association Rule

Let $I = \{i_1, i_2, ..., i_m\}$ be a set of m distinct attributes, also called items. Each transaction T in the database D of transactions has a unique identifier, called TID, and contains a set of items, such that $T \subseteq I$. An association rule is an implication of the form $X \Rightarrow Y$, where $X \subset I, Y \subset I$, are sets of items called itemsets, and $X \bigcap Y = \Phi$.

The rule $X \Rightarrow Y$ holds in the transaction set D (X is called body, Y is head) with support s, where s is the percentage of transactions in D that contain $X \bigcup Y$, i.e., $support\ (X \Rightarrow Y) = \dfrac{\left|\{T : X \bigcup Y \subseteq T, T \in D\}\right|}{\left|D\right|}$. The rule $X \Rightarrow Y$ has confidence c if c is the percentage of the transactions containing X also contain Y, that is, $confidence\ (X \Rightarrow Y) = \dfrac{\left|\{T : X \bigcup Y \subseteq T, T \in D\}\right|}{\left|\{T : X \subseteq T, T \in D\}\right|}$ or $c = \dfrac{Support\ (X \bigcup Y)}{Support(X)}$.

If we do not consider the support and confidence value, there will exist infinite association rules in transactional database. Typically association rules are considered interesting if they satisfy both a minimum support threshold and a minimum confidence threshold. Rules that satisfy both a minimum support (min_sup) threshold and a minimum confidence (min_conf) threshold are called strong association rules. The former denotes item set meet the lowest degree in statistical meaning, i.e., the correctness of rule; the latter reflects the minimum reliability.

The task for association rule mining is to reveal the rules that satisfy given minimum support threshold and minimum confidence threshold. The task can be broken into two steps. The first step consists of finding all large itemsets: each of these itemsets will occur at least as frequently as a pre-determined minimum support value. The second step consists of generating strong association rules with a user specified confidence among the large itemsets found in the first step.

An extensive association rule is often put into use because concept hierarchies are always existed in itemset, therefore has higher support and confidence value. For example, jacket and ski pants belong to out wear, out ware and shirt to more general concept clothes. Having hierarchy relation, more meaningful rules can be found. For instance, buy coat together buy shoes. Here coat and shoes are concept in higher level, so the rule is an extensive association rule. It is difficult to discover useful rule in the market which have thousands upon thousands of items for the lower support count for each item. If higher concept level item considered, the more useful rules will be found.

2.2 Approaches to Mining Association Rule

The association rule mining is brought attention by the experts of artificial intelligence and database. Many algorithms have been proposed in the literature. Similar to the generic data mining process, association rule mining is composed of three

phases. The first phase is to preprocess data, manipulating database and forming transactional database. The second phase, a kernel of whole, is to find all large itemsets that meet the minimum support count. The last phase is to output and interpret association rules that meet user-specified minimum confidence value.

2.3 The Study of Association Rule Mining Algorithm

The Apriori [1] is an influential algorithm for association rule mining proposed by Agrawal. Apriori employs an iterative approach to scan the transactional database several times. It based on the theory that in transactional database D, all subset of large itemset is large itemset, and any superset of weak-itemset is weak-itemset. During the initial pass over the database the support for all single items 1-itemsets L_1 is found. The large 1-itemsets are used to generate candidate 2-itemsets. The database is scanned again to obtain occurrence counts for the candidates, and the large 2-itemsets are selected for the next pass. This iteration process is repeated for k=3,4,…, until there are no more large k-itemsets to be found. The characteristic of Apriori is efficient for small candidate itemsets. For a large transactional database, having great k, Apriori needs k scans to spend more time in I/O, so the efficiency is lower.

Many data mining algorithm [3][6][7] have been proposed that focus on improving the efficiency by reducing candidate set and scan times. For example, the Apriori Hybrid algorithm is put forward by R.Agrawal; hash-based technique presented by Park, which can be used to reduce the size of the candidate k-itemsets; Savasere adopted partition technique to require just two database scans to mine the frequent itemsets; the basic idea of sampling approach (H.Toivonen) is to pick a random sample of the given data. There are other variations involving the mining of distributed association rules such as FDM by D.W.Cheung. [5]

3. THE IMPROVED ALGORITHM BASED ON FRONTIER SET

For improving the efficiency of Apriori, this paper introduces frontier set concept. The frontier set is an itemset ready for candidate itemset. It is called prediction extension algorithm for prediction reason. The basic idea is that to predict each item in extension itemset according to some rule, if the item is frequent itemset then continue to extend otherwise stop extension. After finishing scan, start a next scan for frontier set, which composed of predicted infrequent and belongs to frequent itemset practically. If the rule is chosen better, the result will be closer to the real situation. It's advantage is to reduce the size of candidate set, the scan times of transaction database can be reduced too.

3.1 Introducing of Frontier Set

When the size of transactional database is bigger, there exists statistical principle as follows.

Let $I=\{i_1, i_2, \cdots, i_m\}$ be a set of m distinct items. Each element of frontier set is an item of I. For an element $X+Y$ of candidate set, while X belongs to frontier set and $Y=\{i_1, i_2, ..., i_m\}$. That is to say $X+Y$ is the first extension of X. X appears x times in all transactions, while $X+Y$ appears first time and the X appears c times.

The expectation support of $X+Y$ is:

$$s' = \frac{\int(I_1)\times\int(I_2)\times\cdots\int(I_k)\times(x-c)}{|D|}.$$

The expectation support s' is closer to support count s while transactional database D contains more.

According to the statistical principle we can decide whether frontier set will be entered into candidate set. Like that candidate set is the precondition of large itemset in Apriori, frontier set is the precondition of candidate set in this paper. Through handpicking candidate set, the algorithm improves efficiently. The improved algorithm is broken into following procedures.

3.2 Preprocess

The task of this process is to count the frequency of each item appeared in the transactional database. It belongs to first scan and counts support value for each item.

```
Procedure1 FREQ ( )
Begin
C= ϕ ;              /* C is candidate */
For each transaction t in the dataset D do
{
        if candidates Ci contained in C
           then     i.count ++
        else  {
                    i.count = 1;
                    C = C + Ci;
}
     }
End
```

3.3 Generate Large Itemset

Procedure 2 generates large itemset through scanning transactional database several times. The scan operation is repeated until the frontier set is empty.

```
Procedure2 large_item ();
Begin
L= ϕ ;                  /* L is large itemset */
F= { ϕ };       /* F is frontier set */
when  F≠ ϕ  do
{
C= ϕ ;                  /* C is candidate set */
for all transactions t in the D do
        for all elements f in the F do
        {
          if f ⊆ t  then EXTEND（f.count, f, t）;
if  f ∈ C  then f.count=f.count+1
        else {
                f.count = 1;
                C= C + f ;
  }
        }
        F = ϕ ;
    for all candidates c in the C do
    {
if  count(c)/|D| > minsup   then   L = L + c
else   F = F + c;
    }
}
End
```

After each scan, compare the support count with the minimum support threshold for item in candidate set to determine which one is in large itemset, and whether be added into frontier set in the next scan. If the item is added into large itemset or frontier set, the support value of item is retained.

3.4 Extending of Frontier Set

Procedure 3 is the core of the improved algorithm.

```
Procedure3 EXTEND(int xp，F，t)
```

Begin
I_j = maximum number of frontier set;
for all item number I_k larger than I_j
{
if $I_k \in C$ /* I_k is candidate, count */
if I_k.count > 0 then I_k.count ++
else I_k.count –1
else /* add into candidate set */
{
C = C+ I_k ;

$$p = \frac{I_k.count}{|D|} \times \frac{x_p - x.count}{|D|}$$

if p > minsup then
{
EXTEND(p, I_k, t)
I_k.count = 1
}
else I_k.count = -1
}
}
End

3.5 Generate Association Rules
Procedure4 RULE(L)
Begin
for all candidates I_k in the L do
{
if $\frac{L.count}{(L - I_k).count} > minconf$ then

display($(L - I_k) \Rightarrow I_k / \frac{L.count}{|D|} / \frac{L.count}{(L - I_k).count}$);
}
End
Association rules can be generated easily.

3.6 The Summary of Improved Algorithm
The difference between Apriori and this algorithm is: the former generates candidate set with length k after the k-th scanning transaction database, it spends more time in input and output and leavens the practical use; the latter reduces the size of candidate set and the times of scan and enhances the efficiency.

4. APPLICATIONS

We apply the improved algorithm in the MIS of medical company, which covers the transaction data of stock, sell and storage. The MIS has been running for several years and accumulates considerable sales data. It provides condition and environment for the running of algorithm. Meanwhile customer are not satisfied with ordinary processing, they wanted to assistant decision making utilizing existed data and transition to electronic commerce further. For this reason we use the improved algorithm to analyze the sales data of medical company, only in the sales statement accounts at present.
The sale statements table (yy_xsmxz) of a branch company is used, the data column are sale serial number (xsdjh), coding of merchandise (spbm) and sales date (xtxsrq). For obtaining the data source, take project operation on the sales statements account table over these three attributes. Let serial number as TID, code as transaction item, date as mining range. Association rule mining is carried out in two levels: a product and a kind of merchandise. Some valuable association rules via setting minimum support value and minimum confidence value are uncovered.
We developed an association rule storage system using PowerBuilder and Sybase SQL Server. An association rule base (sales) is designed and has three relation tables: sales_body, sales_head and sales_para. The basic association rule function for sales data is realized. The graphic user interface is provided in the system, the history parameter record curve of an association rule is displayed. Customer examines change process through choosing different parameters.
Candidate set and frontier set will take up much memory room when dealing with large-scale transactional database, so the occurrence of lack of memory may be encountered. For ensuring the successes, it is necessary to manage the memory efficiently. There are two approaches to get more memory in application. One is to write the intending frontier set into disk; the other is to delete some candidate set.

5. CONCLUSIONS

People are not satisfied with computing transactional process and hope to do market basket analysis, such as which groups or sets of items are customers likely to purchase on a given trip to the store. The results may be used to plan market strategies, store layouts, as well as business intelligence. This paper improved existing mining algorithm of association rules. Discovering association rules from transactional database is very important for decision-making in retail market and other business activities.

6. REFERENCES

[1] Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: Proc of the 20th Int'l Conf on Very Large Databases. Santiago, Chile, 1994. 478-499
[2] M. Chen, J. Han, and P. Yu. Data mining: An overview from database perspective. IEEE Transactions on Knowledge and Data Engineer 8(6):866-883, 1996
[3] Jiawei Han, Micheline Kambr, DATA MINING Concepts and Technique. High Education Press, 2001.5
[4] R. Agrawal, T. Imielinski, and A.swami. Database mining: A performance perspective. IEEE Trans. On Knowledge and Data Engineering, 5:914-925, 1993
[5] S.D.Lee, D.W. Cheung, and B. Kao, Is sampling useful in data mining? a case in the maintenance of discovered association rules. Data Mining and Knowledge Discovery, 2(3):233-262, 1998
[6] Tie zhixin, Chen qi, the summarize of association rule, Computer Application Research, 2000.1
[7] Hu kan, Xia Shaowei, the data mining based on great data warehouse: research summarize, Software transaction, 1998.1

Distributed Implementation of Genetic Algorithm to Solve Multiple Traveling Salesmen Problem

Shengping Jin
Department of Statistics, Wuhan University of Technology
Wuhan, Hubei Province, 430063, P.R. China
Intell. Information Process Lab. of Inst. Computing Tech. Academia Sinica
Beijing, 100080, P.R. China
E-mail: spjin@public.wh.hb.cn

ABSTRACT

In Traveling Salesman Problem, given the number of cities and the distances between them, the task is to find the minimum-length closed tour that visits each city once and returns to the starting city. Many applications are involved with multiple salesmen. Each salesman visits a subgroup cities and returns the same starting city. The length of all tours are required to be minimum. This is called Multiple Traveling Salesmen Problem (MTSP).
There are various heuristic approaches to obtain optimal or near-optimal solutions for the TSP problems. But to the Multiple Traveling Salesmen Problem, there are not much approaches to research into MTSP.
In this paper, a distributed implementation of genetic algorithm to solve Multiple Traveling Salesmen Problem is presented. This algorithm combines GA and heuristics. Using island model, the distributed algorithm is constructed in the environment based on network of workstations. Numerical experiments show that the new algorithm is very efficient and effective.

Keywords: Distributed Computation, Genetic Algorithm, TSP, Network of Workstations.

1. INTRODUCTION

The traveling salesman problem, or TSP for short, is easy to state: given a finite number of "cities" along with the cost of travel between each pair of them, find the cheapest way of visiting all the cities and returning to your starting point. The travel costs are symmetric in the sense that traveling from city i to city j costs just as much as traveling from j to i. Many applications are involved with multiple salesmen. Each salesman visits a subgroup cities and returns the same starting city. The sum of all length of each tour is required to be minimum. This is called Multiple Traveling Salesmen Problem (MTSP).
The Traveling Salesman Problem (TSP) falls into the category of NP-complete problems. So does the Multiple Traveling Salesmen Problem. These problems are considered to have no solution in real time. However, there are many ways of approximating the solutions to these problems within specified tolerances. Thus, arbitrarily close solutions may be obtained. There are various heuristic approaches to obtaining optimal or near-optimal solutions for the TSP problems[1,2]. But to the Multiple Traveling Salesmen Problem, there are not much approaches that exist in solving MTSP.
The Genetic Algorithm (GA) is an optimizing algorithm that is modeled after the evolution of organisms. Hybrid methods that combine GA with other techniques have been attempted. A MTSP solver in this paper uses GA and the 2opt method. Owing to the 2-opt, it is much faster than other TSP solvers based on GA alone.
Before proceeding further we need the following definitions:
k-Opt Move is a local heuristic move of swapping k edges in a given tour with edges not in the tour, to generate a new resulting tour. 2-opt move is often used.
GSX is Greedy Subtour Crossover. The solution can pop up from local minima more effectively than by using simulated annealing (SA) methods. In the GSX, we use the path representation for genetic coding.

2. A MTSP REPRESENT

Suppose a MTSP has N cities and M salesmen, the cities are coded by $0,1,\cdots,N$-1. M salesmen start from city 0, visit some cities and come back to city 0. In order to apply algorithms of TSP for MTSP, M-1 virtual cities coded by N, N+1,$\cdots$,N+M-2 are appended to the N cities. The distance between city i($0<i<N$) and city j($N\leqslant j<N+M$-1) is defined by the distance between city i and city 0. The distance between city i(($N\leqslant i<N+M$-1)) and city j($N\leqslant j<N+M$-1,$i\neq j$), or the distance between city 0 and city j($N\leqslant j<N+M$-1) is infinite for easy computation implementation.
A natural candidate for a tour data structure is to keep an array, called tour, of the cities in the order they appear in the tour. To locate a given city in tour, we use another array, called Inv(Inverse), where the ith item in Inv contains the location of city i in tour. The two arrays in Figure 1 illustrate the data structure for the tour 9-0-8-5-7-2-6-1-4-3.

Tour	9	0	8	5	7	2	6	1	4	3
Inv	1	7	5	9	8	3	6	4	2	0

Figure 1 Array for tour data

Figure 2 represents for a 10 cities and 2 salesmen MTSP. The two salesmen visit two subtours 0-9-1-8-5-3-0 and 0-7-6-2-4-0. City 10 in Tour in Figure 2 is a virtual city.

Tour	0	9	1	8	5	3	10	7	6	2	4
Inv	0	2	9	5	10	4	8	7	3	1	6

Figure 2 Represent of a MTSP

Owing to the distances between virtual cities and starting city are infinite, the algorithms to solve TSP are also suitable to MTSP in general. However, some extra requirement is need for a MTSP. For example, the visiting distances of salesmen are wanted approximate equal. So the genetic algorithm to solve a TSP is much better than other algorithms to solve a

MTSP because the fitness function is easy to modify to guide the evolution direction. This is showed in numerical example in section 5.

3. HEURISICS AND GA FOR TSP

The genetic algorithm (GA) is a model of machine learning based on biological evolution of population of individuals. The processes of evolution in nature seems to boil down to different individuals competing for resources in the environment. Some are better competitors than others. Those that are better are more likely to survive and propagate their genetic material. The selection of individuals over generations is mainly determined by what we call genetic operators. There are two basic operators that are commonly used to generate offspring in the next generation: **crossover** and **mutation.**

For TSP, to find the shortest path in a traveling salesman problem, each solution would be a path. The length of the path could be expressed as a number, which would serve as the solution's fitness.

We present a hybrid method that uses GA and the 2opt method. In our GA, the 2opt method provides mutation, while the crossover operator provides the capability of jumping out from the local minima, where the solution often falls when only 2opt is used. The algorithm consists of the following steps.

Initialization: Generation of Scale individuals randomly.

Multiplication crossover: Choose Scale × pe% pairs of individuals randomly and produce an offspring from each pair of individuals. The population reverts to the initial population Scale.

Mutation by 2opt: Choose all individuals and improve them by the 2opt method at beginning. Then each offspring produced by crossover is applied by 2opt.

We now describe the detail of two key steps.

3.1 Mutation by 2opt

The 2opt method is one of the most well-known local search algorithms among TSP solving algorithms. It improves the tour edge by edge and reverses the order of the subtour.

We repeat the procedures described above until no further improvement can be made.

For an N-city TSP, any optimal tour, $(i_0,i_1,\cdots,i_{N-1})$, for each $0 \leqslant p<q<N$(subscript will be taken modulo N) we have

$$w(i_{p-1},i_p)+w(i_q,i_{q+1})< w(i_{p-1},i_q)+w(i_{p-1},i_{q+1}) \quad (1)$$

$w(i_{p-1},i_p)$ in (1) denotes the distance between city i_{p-1} and city i_p. A pair (p,q) that violates (1) is called an "intersecting pair", and 2opt method is to fix any such pair until the tour become "intersectionless".

The observation is that if (p,q) intersect in the tour

$$(i_0, \cdots, i_{p-1}, i_p, \cdots, i_q, i_{q+1}, \cdots, i_{N-1}),$$

then the pair can be fix by inverting the subsequence $(i_p, \cdots, i_q)$, that is, moving to the tour

$$(i_0, \cdots, i_{p-1}, i_q, i_{q-1}, \cdots, i_{p+1}, i_p, i_{q+1}, \cdots, i_{N-1}).$$

We call this operation flip(i_p,i_q). Figure 3 shows the flip(5,6) of the tour in Figure 1.

To reduce the running time, if p and q are separated from each other by many cities($|p-q|> N/2$), flip(i_p,i_q) can carry on inverse order, Figure 4 shows the flip(8,4) of the tour in Figure 3 on inverse order.

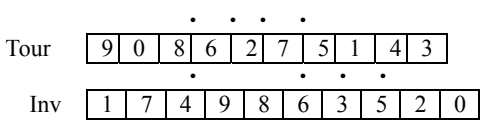


Figure 3 Flip(5,6) of Figure 1

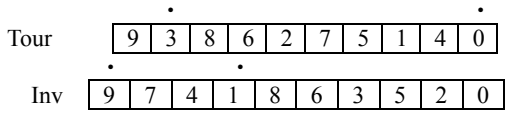


Figure 4 Flip (8,4) of Figure 3 on inverse order

3.2 Crossover in Multiplication

The crossover operation happens in an environment where the selection of who gets to mate is a function of the fitness of the individual, i.e., how good the individual is at competing in its environment. Two individuals selected based on the fitness function are required to crossover to generate an offspring, which may be selected for the next generation.

Greedy Subtour Crossover (GSX) is proposed. The algorithm of GSX is as following:

Inputs: Chromosomes $g_a = (a_0,a_1, \cdots ,a_{N-1})$ and $g_b = (b_0,b_1,\cdots,b_{N-1})$.
Outputs: The offspring chromosome g.
procedure crossover(g_a,g_b) {
f_a=true;
f_b=true;
choose town t randomly;
choose x, where $a_x = t$;
choose y, where $b_y = t$;
g=t
do{
x=x-1 (mod N);
y=y+1 (mod N).
if f_a == true then{
if $a_x \notin$ g then { g=a_x • g;}
else {f_a =false;}
}
if f_b == true then{
if $b_y \notin$ g then { g= g • b_y;}
else {f_b =false;}
}
} while(f_a==true or f_b==true)
add the rest of cities to g in the random order
return g
}

Note that " a_x • g " is the concatenation operator, and that sentence means to add a_x before the chromosome g. "g • b_y " means add b_y after the chromosome g.

4. DISTRIBUTED IMPLEMENTATION OF GA

The algorithm of heuristics and GA for TSP is as follows:

create initial population with random
for each individual in population do 2opt
repeat
  for each crossover do
    select two individuals in population at random
    Apply GSX crossover on these two individuals
    Apply 2opt on the offspring
    with pre-defined probability do mutation on the offspring
    replace an individual in population with by offspring
  done
until converged

The way in which (parallel genetic algorithm) can be implemented depends on the following elements:

- How fitness is evaluated and mutation is applied
- If single or multiple subpopulations are used
- If multiple populations are used, how individuals are exchanged
- How selection is applied (globally or locally)

In the "island" approach to parallelization of genetic programming, the population for a given run is divided into semi-isolated subpopulations. Each subpopulation is assigned to a separate processor of parallel system. 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[3].

5. NUMERICAL EXAMPLE

Figure 5 represents a map of a county. Three inspectors start from the capital of the county and visit all towns and villages, and come back the capital at last. It is required to design three tours, the total distance of which is minimum and three distances are as balanced as possible.

Define: call $\dfrac{\max\limits_{i,j}|w(C_i)-w(C_j)|}{\max\limits_{i} w(C_i)}$ balance measure, where C_i is the i'th tour and $w(C_i)$ is the distance of C_i.

Zhao Jing and Dan Qi[4] get the results as table 1 by using intuitable observing:

Table 1 three tours distances and balance measure

Dist. of Tour 1	Dist. Of Tour 2	Dist. Of Tour 3	Total dist.	Balance measure
191.1km	216.4km	192.3km	599.8km	11.69%

Considering the balance measure, by slight modifying the MTSP solver algorithm, we can get the three tours which is better than the result as above. Table 2 shows the detail of the new result.

The cities in parentheses in table 2 are passed only but not visited. The balance measure of the new tours is 10.20%, which is smaller than 11.69%. The total distance is reduced by 2.4km.

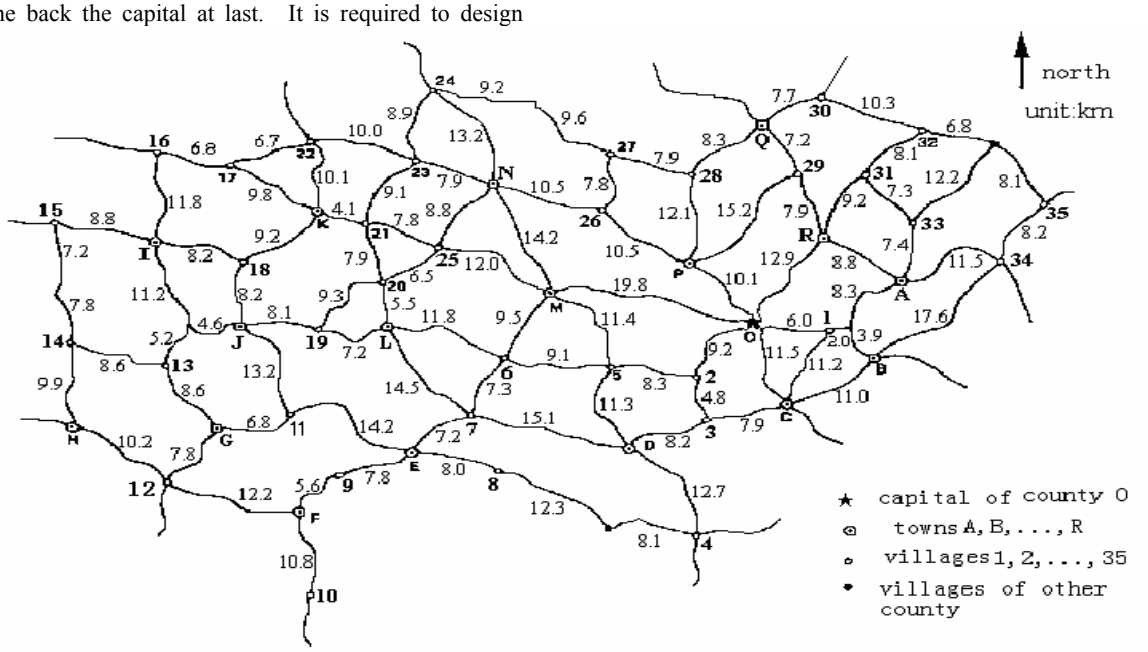


Figure 5 Map of a County

Table 2 Detail of the new three tours

No.	Tours	Dist. of Tours	Total distance
1	O-(P-26-)N-24-23-22-17-16-I-15-(I-)18-J-19-L-20-21-K-(21-)25-M-O	199.0km	597.4km
2	O-5-6-7-E-9-F-10-(F-)12-H-14-13-G-11-(E-)8-4-D-3-2-O	209.9km	
3	O-R-A-34-35-33-31-32-30-Q-29-(Q-)28-27-26-P-(O-)C-B-1-O	188.5km	

6. REFERENCES

[1] Sengoku, H., Yoshihara, I., A Fast TSP Solution using Genetic Algorithm (Japanese), Information Processing Society of Japan 46th Nat'l Conv. 1993

[2] Martina Gorges-Schleuter, ASPARAGOS: An asynchronous parallel genetic optimization strategy, In Proc. 3rd ICGA, Morgan Kaufmann Publishers, 1989, 422-427

[3] Xiong Shengwu, Chu Wujun, NOW-Based Distributed Implementation of Evolutionary Algorithms. DCABES 2001 Proc., Hubei Science and Technology Press, 2001,40-43

[4] Zhao Jing, Dan Qi, Mathematical Modeling and Mathematical Experiments, Beijing: China Higher Education Press and Springer-Verlag, 2000, 151-159

A Novel Algorithm for Automatic Rectangular Meshing

Shuxuan Shen and Yaming Bo
Communication and Control Engineering Institute, Southern Yangtze University
Wuxi, Jiangsu province, China
E-mail:shen_sx@sina.com

ABSTRACT

The efficient preconditioner derived from finite difference equations, which is suitable for large-scale computing operations and for solving elliptic differential equation, requires rectangular meshes. By means of the point-by-point comparison method, a new method is proposed, which tremendously simplifies the computer operations and data processing. Based on this new method, rectangular mesh generator is presented in this paper.

Keyword: Automatic Mesh, Point-by-Point Comparison method, Finite difference Equations, Preconditioner, Rectangle Mesh,

1. INTRODUCTION

As a part of a CAD tool for science computation, an automatic mesh generator plays an important role for the calculation efficiency of calculation and the accuracy of result. Most of automatic mesh generators are designed for finite-element solver, are widely used in engineering[1].
Recently, a sparse preconditioner derived from finite difference equation, is superior in comparing to the preconditioners of imcomplete LU (ILU) and modified ILU (MILU) decomposition. [2] The solver with this preconditioner, which is more suitable for large-scale computing and for solving elliptic differential equation, requires rectangular meshes. The commonly used finite-element mesh generators are hard to satisfy this requirement because of irregular mesh shape. By a point-by-point comparison method, a new method is proposed, with which not only rectangular meshes can be generated and refined, but also the computer operations and the data size can be reduced during the geometrical dividing and the data recording processes.
In this paper, at first, the conventional point-by-point comparison method is modified with an approximate processing. Secondly, the mesh refinement is briefly discussed. Then, the results of several mesh generators are compared with each other. Finally conclusion and related.

2. MESHING ALGORITHM

The function of mesh generator is to divide the target area into rectangular meshes with expected mesh sizes according to the complexity of problems. The target areas may have irregular boundaries for practical problems, so an approximate processing on the boundaries is needed. Therefore, the first step of our method is to mesh the region along the boundary. Then the values of mesh elements inside the border can be easily obtained according to the given sizes and discrete values on the boundary.
In order to mesh, a coordinate should be set up to locate the target area, whose axes overlap parts of the target area's boundary if possible. The more they overlap, the less calculation required.

The modified point-by-point comparison method
Because of irregular shapes of the target area, there are mesh elements, which has one part inside the border and another outside. The modified method studies such meshing method. In point-by-point comparison method, if one of these mesh elements is ignored, the following one should be included in the discrete region. The mesh elements ignored and the mesh elements included should alternate along the discrete boundary. If the gradient of one part of the border is very large or very small, the scheme for alternating may lead to two cases. One is that the mesh elements whose part inside the target area is much smaller than the part outside, is included into the discretization area. The idea is just shown in Fig.1 (a), where the target area is in gray. In unit 1 and 2, the parts inside the target area are all much smaller than the white ones, so they should be ignored totally. But one of them should be included into target area, according to the alternative rules[3].The other case is that the mesh elements whose part inside the target area is much bigger than the part outside, is ignored. It is shown in Fig.1 (b). Unit 3 and 4 are better to be included into the target area, but one of them is ignored in the point-by-point comparison method. Therefore the precision of the point-by-pint comparison method is limited because of the problem.

(a) (b)

Fig.1 the cases limiting precision

In order to improve the precision, the method is modified as follow, if the part inside the boundary occupies more than half of the unit, the unit should be included. Otherwise, it should be ignored.
The modified method is based on the hypothesis that, when the unit is small enough, the curve in one mesh unit can be approximated as a line. It is shown in Fig.2 that the region bounded by line *l* and the mesh unit boundary can be approximately considered as the region bounded by the mesh elements borders and the curve c.

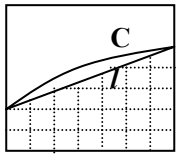


Fig.2 unit on the boundary

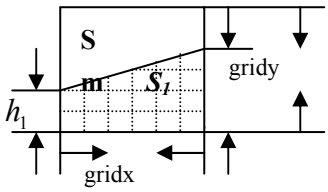


Fig. 3 unit to approximate

In Fig. 3, the area s_1 in shadow in the mesh unit is

$$s_1 = \frac{1}{2}(h_1 + h_2) * gridx \cdot \quad (2)$$

If

$$s_1 < \frac{1}{2}s \quad , \quad (3)$$

then

$$\frac{1}{2}(h_1 + h_2) * gridx < \frac{1}{2} gridx * gridy \cdot \quad (4)$$

Eq（4）is equivalent to

$$h_1 + h_2 < gridy \cdot \quad (5)$$

According to 2-point line formula, the line m can be represented as

$$y = \frac{h_2 - h_1}{gridx} * x + h_1 = f(x) \quad \cdot \quad (6)$$

Then

$$f(\frac{1}{2}gridx) = \frac{h_2 - h_1}{gridx} * \frac{1}{2}gridx + h_1 = \frac{1}{2}(h_1 + h_2) \cdot \quad (7)$$

From Eq（5）and Eq（7）, we have

$$f(\frac{1}{2}gridx) < \frac{1}{2}gridy \quad \cdot \quad (8)$$

And when $s_1 < \frac{1}{2}s$,

$$f(\frac{1}{2}gridx + x_i) < \frac{1}{2}gridy + y_i \quad \cdot \quad (9)$$

On the contrary, when $f(\frac{1}{2}gridx + x_i) < \frac{1}{2}gridy + y_i$,

$$s_1 < \frac{1}{2}s \quad (10)$$

Therefore, we can judge whether to include or to ignore the units on the boundary by comparing the values of $f(\frac{1}{2}gridx + x_i)$ and $\frac{1}{2}gridy + y_i$.

The procedure to move along the boundary is as follows:

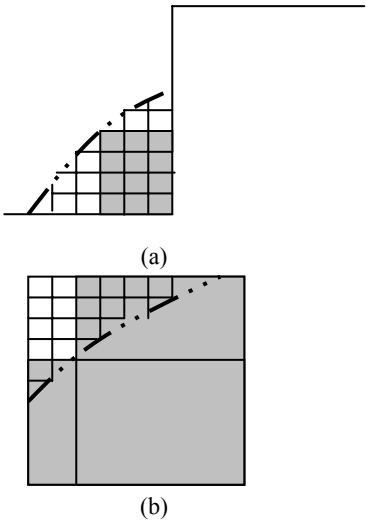
1) select one axis as the main axis to move.
2) Move one step according to mesh size along the main axis.
3) Then, calculate the values of $f(\frac{1}{2}gridx + x_i)$ and $\frac{1}{2}gridy + y_i$ to judge how to make an approximation.
4) If units need to be included, move one step to the next point along the other axes, else do nothing.
5) Return to step 2) until it returns to the start point of the boundary.

This method can be referred to the point-by-point comparison method.[3]

3. MESH REFINEMENT

Sometimes, in cases that the target region possesses complex geometrical structures or required function has a sharp distribution in part of the region, higher accuracy of the result may be required. In order to satisfy this requirement, fine mesh is necessary.

Mesh refinement is to divide the unit into smaller subunits according to the requirement of users. The manipulation is similar to the modified point-by-point comparison method. Firstly, the user should select the area to be refined and set mesh steps for subunits. The original unit in that area will be divided into several subunits according to the mesh step values. For the units completely inside the border, the procedure can be referred as the automatic meshing method. However, the units on the boundary should be dealt with separately as shown below, according to the original boundary lines.



(a)

(b)

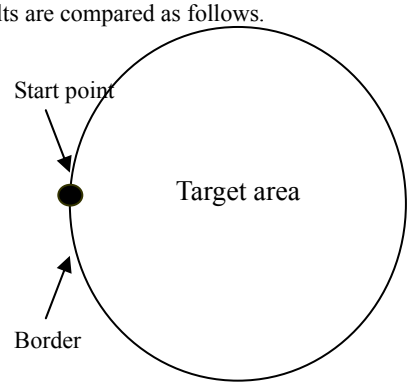
Fig. 4 unit to mesh refine

There are two cases after the approximation in automatic meshing, as shown in Fig.4. In Fig.4 (a), a part of target area with shadow under the broken line, which is smaller than half of unit, is ignored in automatic meshing described in section 2. But it is big enough to form a new subunit during mesh refining. So a new subunit with shadow in gray should be added to the original region. As a result, smaller area with shadow in white is ignored. In Fig.4 (b), the area without shadow occupies more than half of unit, which is totally added into the target area during the automatic meshing. It should be calculated again for better approximation in mesh refinement. As shown in Fig.4 (b), the smaller area with shadow in gray is included.

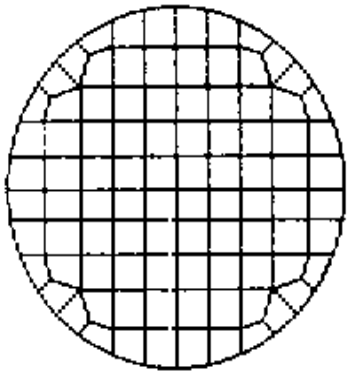
This treatment is the key to improve the precision in mesh refinement. Furthermore, the smaller the unit, the higher the precision. However, the trade-off between precision of the solution and the number of mesh nodes mainly depends on the users experience.

4. COMPARISON TO RELATED WORK

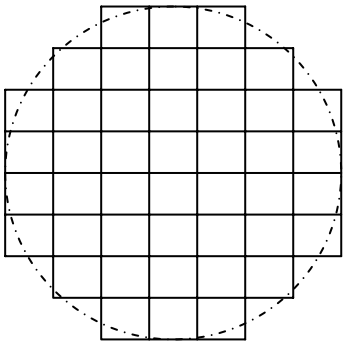
A circle is meshed with finite-element mesh generator and the new rectangular mesh generator presented by authors. The results are compared as follows.



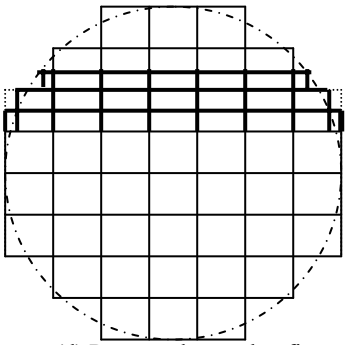
(a) The original target area



(b) Finite-element mesh



(c) Rectangular mesh



(d) Rectangular mesh refinement

Fig .5 mesh from different generator

In Fig.5, mesh (b) comes from finite-element mesh generator, (c) comes from rectangular mesh generator, and (d) is the result of mesh refinement after rectangular meshing.

It can be seen that finite-element mesh generator make the discretization strictly according to the boundary of target area, whereas rectangular mesh generator is not so accurate. Obviously, the former is superior in the precision of the discretized boundary to the latter.

On the other hand, with rectangular mesh, the values of units inside the border can be easily obtained by rectangular mesh generator. The finite-element mesh generator must calculate and record these values inside its own target border. Therefore, rectangular mesh generator is more efficient in memory space and time-consumption for meshing.

In comparison Fig.5 (b) with (c), the latter is more accurate according to the target region. So mesh refinement can improve the precision of meshes.

5. CONCLUSIONS AND FUTURE WORK

The rectangular mesh generator justly serves the finite difference solver. The computer operations and the data size can be reduced during the geometrical dividing and data recording. Although the limited border precision compared with finite-element mesh generators is its flaw, the rectangular meshes generator meet the needs of the finite difference solver. In order to improve precision of meshes, reducing the size of unit mesh and mesh refinement are good schemes.

The rectangular mesh generator has been implemented for CAD software with a solver of finite-difference and conjugate gradient method (FDM).

If the target area is very complex, it will take a long time to mesh. In order to save the time to mesh, this target area can be divided into several sub-areas. Then these sub-areas can be meshed separately on parallel computers. This is our future work.

6. REFERENCE

[1] Shi xiaoxiang, “A object_oriented technique applied in Finite Element meshing system”, Journal of computer application and research, vol.16, 2000, pp37-38

[2] Yaming Bo et al. A Preconditioner Derived from Finite Difference Equations for Solving Poisson’s Equation, 2001 International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2001 PROCEEDINGS, Hubei: Hubei Science and Technology Press, Wuhan, China 2001，pp28

[3] Xie JianYing, “Microcomputer control technique”, Beijing: Shanghai transportation university pub, 1998, pp123-132

Solving TSP with Distributed Genetic Algorithm and CORBA

Yijiao Yu, Qin Liu and Liansheng Tan
Department of Computer Science, Central China Normal University, Wuhan 430079, PR China.
E-mail: {yjyu, liuqin, L.Tan}@ccnu.edu.cn

ABSTRACT

Distributed Genetic Algorithm (DGA) has been used to parallel computing and Common Object Request Broker Architecture (CORBA) is popular in software integration. In order to reduce the complexity of DGA software development and to minimize the maintenance costs, DGA software with CORBA and Java (DGAandCORBA) is proposed. The advantages and software architecture of DGAandCORBA are illustrated in detail. China Traveling Salesman Problem (CTSP) has been computed with different parameters and three rules about parameters selections are exposed. It's shown that the solution quality with best-migration policy is the same as that of random-migration policy, the good solution is easily available both in the case when migration rate is about 20‰ and reproduction generation is between 25,000 and 30,000. Finally, Several experiments are carried out to verify these rules. Experiments show the efficiency of our approach in solving TSP.

Key words: TSP, CORBA, Java, Genetic algorithm, DGAandCORBA.

1. INTRODUCTION

Traveling Salesman Problem (TSP) is well known as a NP-hard problem. TSP is used to test novel algorithms in computer science due to its simple description. The problem is that there're *N (N>0)* cities in a graph, and a salesman will visit all the cities but every city should be traveled only once except the beginning one. Which traveling sequence will lead to the shortest distance? Obviously, we can look on the graph as a map and every city has its coordinate (x_i y_i). The distance of the traveling loop can thus be calculated with equation (1) and the distance between two different cities can be calculated with equation (2). According to arranging principle, we know there're $\frac{N!}{2N}$ elements (T_d) in the solution space.

$$T_d = \sum_{i=1}^{N-1} d(v_i, v_{i+1}) + d(v_N, v_1) \quad (1)$$

$$d(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (2)$$

Since 1980s, there have been so many algorithms proposed to solve TSP problem, such as Genetic Algorithm (GA), Hopfield Neural Network (HNN), Ant Colony Algorithm. In the past, we have tested and compared some algorithms in experiments and concluded that GA is better than artificial neural network in software designing, testing and converging speed, especially when *N* is more than 20 [1]. As we have observed, DGA software usually runs in three kinds of hardware environments in China. Some DGA programs run on super computers or great clusters, some are still running on PC or workstation, and others run on Local Area Network (LAN) with TCP/IP. For economic reasons, most universities lack super computers and great clusters. On the other hand, software in super computer isn't portable to PC or workstation. Therefore, the first way is limited by the hardware conditions and portable ability. Obviously, the second one is unfit for large scale TSP. Since every university has its own LAN, the third is popular in China. Considering factors above, we decide to solve large scale TSP on LAN. In other countries, there're also discussions about DGA on Internet or Cluster [2][3].

The rest of the paper is organized as follows: Section 2 illustrates the advantages about DGAandCORBA and the 3-layer software architecture; Section 3 presents experiment methods and results; In order to find a good parameter combination, we have tried a great number of combinations, and section 4 shows three rules about parameters combination. In section 5, there're some testing examples to prove that DGAandCORBA is able to find the optimized solution in an acceptable time slot. The paper concludes with section 6.

2. DESIGN DGA WITH CORBA AND JAVA

Why uses CORBA and Java to realize DGA

There're two kinds of way to design communication software, one is TCP/IP socket-based, and the other is CORBA-based. As to the former coding DGA software with TCP/IP socket, first, the server and client will discuss the communication protocols about how to represent the data and how to deal with the transmission error and exception; and then, select the suitable programming language. However, sockets in different languages have different methods; finally, code and test programs again and again. Testing communication software is a hard job and nearly nobody would like to do it. In a word, Designing DGA with sockets is a troublesome work and couldn't get good software maintenance ability.

CORBA is proposed and developed by Object Management Group (OMG), and has become the most popular middleware technology in the world. Compared with COM and DCOM from Microsoft, CORBA is supported by more and more software companies. The most important reason is that Object Request Broker (ORB) in CORBA communicates with Internet Inter-ORB Protocol (IIOP), which makes that calling remote objects as if in the local [4]. So the DGA software designer doesn't need to care about the machine address and communication protocols.

Java has been regarded as the most successful platform which is an independent portable language. Almost all Operation Systems (OS) support it, such as Windows, Unix, Solaris and Linux. Computer networks are becoming larger and more heterogeneous. Even in a LAN, different OSs and machines exist too. In order to make full use of the LAN and insure compiling once and running everywhere, Java is the only language to be chosen.

For the sake of reusing software in future and cutting down the costs of software maintenance, we design DGA with CORBA and Java.

The research is supported by the grant of Key Science & Technology Foundation in Hubei Province (2001AA104A05) and Natural Science Foundation in Hubei Province (2001AA104A05).

The 3-layer software architecture

Fig.1 shows the 3-layer software architecture about DGAandCORBA. The external layer is the Remote Control GUI layer which can run anywhere in Internet. Users can visit the application server with web browser, such as Netscape, Internet Explorer. Users can define a TSP, set parameters and get results without knowing which machines are computing for them. Due to the security and using permission, the internal layer runs in a LAN. But as far as the technology is concerned, it can also run in Internet. There are lots of islands in internal layer. Each island is a software process and a meta computing unit in each machine. The computing ability of single unit is simple and low, but they are parallel, hence the sum of computing speed is considerably high.

The middle layer is the Application Sever, including a WWW Sever and a DGA Sever. WWW Sever is to publish web pages and DGA Sever is to control the DGA computation. When DGA sever accept a computing request from the Remote Control GUI, it will find idle islands from the Java Naming and Directory Interface (JNDI), then pass parameters and assign tasks to islands. DGA sever also needn't to know where the active islands are. If the scale of TSP is very large, a great number of islands to compute are needed, Lightweight Directory Access Protocol (LDAP) Sever will replace the JNDI. In fact, DGA sever does nothing but to control the islands. For the security reason, Java applet is strictly limited to communicate with the host machine only, which is the WWW sever. So DGA sever must run in the same machine with WWW server. To sum up, the DGA sever process needs little memory and CPU resource, so the security request won't bring any extra hardware requirements.

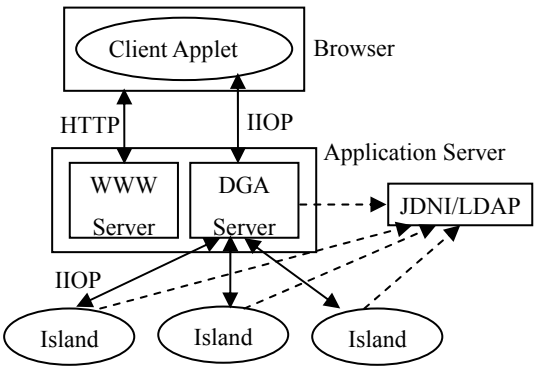


Fig. 1 The 3-layer DGA software architecture.

3. EXPERIMENT METHODS AND RESULTS

There're 34 nodes in CTSP, and every node represents the capital of a province, including Hong Kong, Macao and Taipei. The algorithm in island is the simple genetic algorithm. Fitness is the distance of traveling loop. Different from usual genetic algorithm, chromosomes with small fitness in this experiment will have more living chances. Chromosome coding method is vital in crossing and mutation operation. One dimension array is used to represent chromosome here. The number of arrays is the traveling sequence number and the data in element is the traveled city number. For example, chromosome in diagram 1 represents a traveling sequence "3-2-0-1-3". Simple chromosome coding method is used mainly for two causes. First, the representation is so simple that programmers can easily understand it. Second, it's able to support large scale of TSP. In this case, there're 34 cities in CTSP, so the maximum subscript of array will be 33. If there're 101 cities in TSP, the subscript will increase to 100. All the crossover and mutation operations doesn't need to do any change when the TSP is changed.

3	2	0	1
0	1	2	3

Diagram 1 A chromosome example.

Parent selecting algorithm is divided into 4 steps, namely as below: 1.Select num (num is the league match size) chromosomes from the population arbitrarily; 2. Sort all chromosomes according to their fitness; 3. Select the shortest two to be parents; 4. Cross and produce two children, and the children will replace the worst two in the num chromosomes. We consider crossover is more important than mutation, so with crossover probability set to 0.999, and the mutation probability 0.001.

There're three islands in this computing system. Every island will immigrate some chromosomes to other islands randomly after reproduction times. The migration chromosomes will replace the worst in island.

We get the shortest distance when the parameters combination is as below, population size is 1200 in every island, migration rate is 20‰, and reproduction generation is 25,000. The shortest distance is 15676 km. This one is better than the past resolutions with other methods [5]. The visiting sequence is shown in Fig.2.

Fig. 2 The shortest loop.

4. RULES ABOUT PARAMETERS COMBINATION

Best migration policy VS. random migration policy

Obviously, different migration policy will bring different impact on solution quality. There're two popular kinds of migration policy, such as the best-migration(BM) and random-migration(RM) in DGA. With BM, every island will sort all chromosomes first, then collect the best subset and send them to other islands. It has been proved that the average time complexity of best sorting algorithm is *nlog(n)* (n is the elements number) [6]. Usually there're more than 1,000 chromosomes in an island, so the sorting is time-consuming. On the contrary, RM doesn't need to do that and the computing velocity will subsequently be fast.

It [7] has been pointed out that the solution quality with RM isn't worse than others. In order to verify it, we compute 50 times and get 50 solutions first, for every parameters combination. Then select the shortest 30 ones and accumulate their mean-distance. Fig. 3. shows the shortest and the mean distance under RM or BM when the population size in island is 1200 and the migration rate is from 5‰ to 25‰ with a step of 5‰. Fig. 4. shows them when population size in island is 1500. SubP means the population size in figures in this paper. From the figures, we can see that the influence on mean-distance from RM and BM is very similar. But the shortest-distance is different, RM is a little better. In view of the time cost, we consider RM better than BM. Thus all the migration policy is RM in the later experiments.

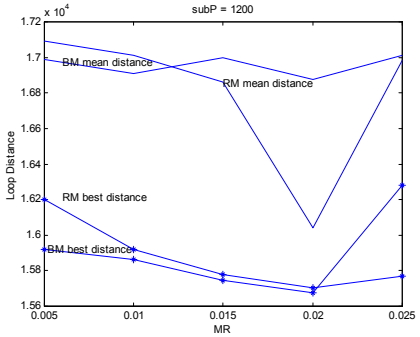


Fig. 3 SubP is 1200,solutions with BM or RM.

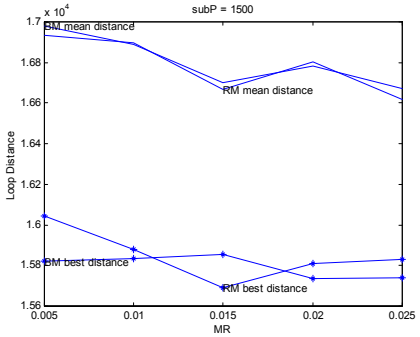


Fig. 4 Subp is 1500,Solution with BM or RM.

Good migration rate is about 20‰

Migration rate (MR) is the percentage of the population moved during each migration in an island. As indicated in some references like [7], MR should not be too high or too low. Too high MR will spend much communication cost and reduce the variety of population, and all islands will easily converge to the same local maximum point. Too low MR will lead to slow convergence speed. To get a good value of MR, we change MR little by little with other parameters value unchanged.

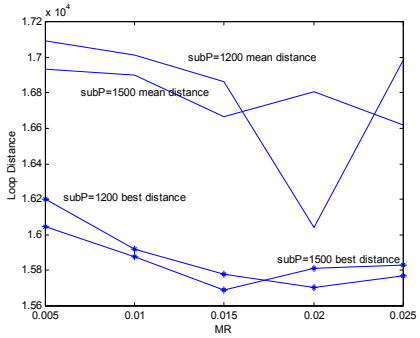


Fig. 5 SubP is 1200, 1500.

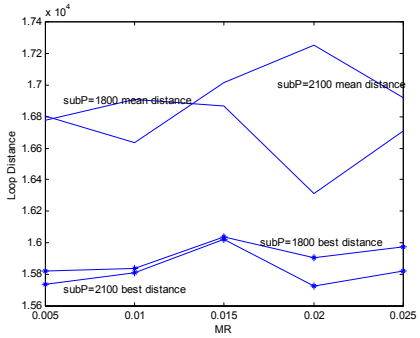


Fig. 6 SubP is 1800, 2100.

Fig. 5. and Fig. 6. show the shortest-distance and mean-distance distributed when MR is from 5‰ to 25‰ and the population size in island is 1200,1500,1800,2100. From the above two figures, we can find that it will soon get the shortest solution when the good MR is about 20‰.

Good reproduction generations is between 25,000 and 30,000

To find a good value of reproduction generation (RG), we increase RG from 5,000 with the step of 5,000. When RG is more than 35,000, the solution quality become stable. That's to say, too large RG can't bring better solution but longer computing time. That's because all chromosomes are converged to one or several local maximum points and new children are also located there. To avoid this phenomenon, we can increase the mutation probability or MR.

Fig. 7. and Fig. 8. show the shortest-distance and mean-distance changes with the RG increasing from 5,000 and 35,000. From the observations, we recommend the number of reproduction generation is between 25,000 and 30,000.

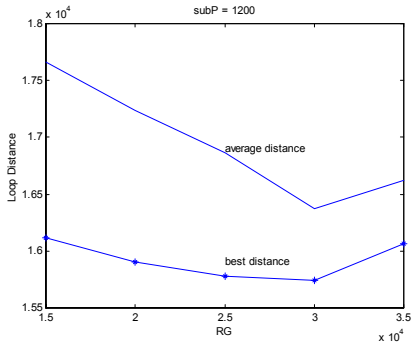


Fig. 7 SubP is 1200 with different RG.

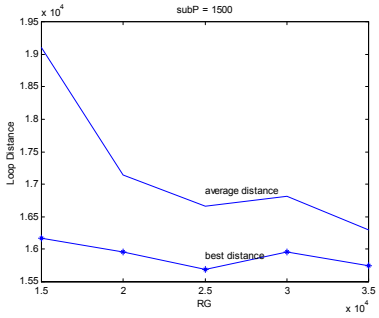


Fig. 8 SubP is 1500 with different RG.

5. DGAandCORBA PERFORMANCE TESTING

In order to test the DGAandCORBA performance, we design 3 special TSPs. The first is 30 points in an rectangle; the second is 40 points in square; and the third is 35 points in a circle. The topology distributed show in Fig. 9. The reasons for designing these 3 TSPs are as below. 1.It's easy to get best solution in math; 2.To test whether DGAandCORBA searching ability depends on points topology. The results are shown in Table 1. They indicate that DGAandCORBA is successful to solve TSP. Comparing with that are reported in [8], it's easy to see that rules about arguments combination are different between DGA and Centralized Genetic Algorithm.

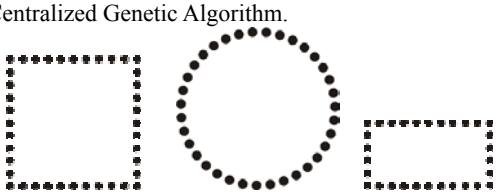


Fig. 9 Testing cases

Table 1 Results for the 3 testing cases.

	M	SE	SM
Rectangle	380	300	300
Square	589	400	400
Circle	422	309	309

M: Mean-distance
SE: Shortest-distance in experiments
SM: Shortest-distance in math

6. CONCLUSIONS

In this paper, we have proposed a method of designing DGA software with Java and CORBA and test it in a local area network. During the computing process, we get three rules about the parameters combination. Because it's difficult to analyze and prove the DGAandCORBA performance in theory, we test them in a number of cases.

Internet is becoming more and more popular, which leads to the increase of free computing resources quickly. Using distributed and cheaper computers to handle complicated science or business computing is the tendency way in future. Grid is just representing this approach and is becoming the next generation Internet computing and application model. From our point of view, application about DGA on LAN and Internet is available for other software computing technology also.

In the future, we should try to analyze the parameter combination rules in a theoretic manner and carry out more testing experiments.

7. REFERENCES

[1] Yu yijiao, "Hopfield Neural Network and Genetic Algorithm in Solving Traveling Salesman Problem: Experimental Comparison and Analysis", Journal of Central China Normal University (Natural science), No.2, pp157-161, 2001

[2] Yusuke Tanimura, "Optimization Methods using a Global Computing Environment", available at http://mikilab.doshisha.ac.jp/dia/monthly/monthly01/20010423/38monthly.

[3] Tomoyuki Hiroyasu, Mitsunori Miki, Masahiro Hamasaki and Yusuke Tanimura, "A New Model of Distributed Genetic Algorithm for Cluster System: Dual Individual DGA", Proceedings of CC-TEA, 2000, available at http://mikilab.doshisha.ac.jp.

[4] CORBA/IIOP2.3.1 Specification, available at http://cgi.omg.org/library/c2indx.html.

[5] Yu yijiao, "The Analysis about parameters in solving TSP with genetic algorithms:", Journal of Central China Normal University (Natural science), No.1, pp25-29, 2002

[6] William Topp. Data structure with C++. Prentice Hall, 1996.

[7] Chen Guoliang, Wang Xifa, Zhuang Zhenquan. Genetic algorithm and its application. People Posts & Telecommunication Publications House, 1999.

[8] Pan Zhengjun, Kang Lishan, Chen Yuping. Evolution Algorithm. Tsinghua University Press and Guangxi Science & Technology Publication House. 1999.

Adaptive Genetic Algorithm For Optimal Distributed Multicast Routing

Youwei Yuan Lamei Yan Xingming Sun
Department of Computer Science and Technology,
Zhuzhou Engineering Institute
Hunan, China
E-mail: yuanyouwei@163.net
M.Mat Deris
Department of Computer Science,
University College of Science & Technology Malaysia
E-mail: mustafa@uct.edu.my

ABSTRACT

Multicast is a mechanism used in communication networks that allows distribution of information from a single source to multiple destinations. The distributed algorithm for a multicast connection set-up, based on Genetic algorithm (GA) is reviewed. GA is a methodology within the artificial intelligence area and it emulates the strive of a species. We present a Genetic algorithm for distributed multicast routing, including building and dynamic maintenance of multicast routing tree in package exchange network. The algorithm has the following characteristics: the preprocessing mechanism, the tree structure coding method, the heuristic crossover technique, and the instructional mutation process. Simulations show that the GA has an optimal network cost to solve the minimal multicast tree with delay constraint. In comparison to other algorithms, it is more efficient and effective.

Keywords: delay constrained, distributed algorithm, genetic algorithm, multicast routing

1. INTRODUCTION

The widespread use of distributed computing system is due to the price-performance revolution in microelectronics, the development of cost-effective and efficient communication subsets, the development of resources sharing software, and the increased user demands for communication, economical sharing of resources, and productivity. The success of a routing algorithm is predicted on having an accurate view of the state of the network, including its topology and the availability of the resources at every node. When the nodes do not have complete knowledge of the topology and state of the network, distributed routing algorithm are needed[11].

Multicast is a mechanism used in communication networks that allows distribution of information from a single source to multiple destinations. Multicast constrained QoS routing optimization is an important issue in the current communication network research. As such, multicast routing algorithms that compute least cost multicast trees under a given end-to -end delay constraint are desirable in order to support some applications efficiently. This is due to the fact that, the memberships of the multicast group change frequently in some applications (e.g. teleconferencing). It should also have the ability to alter an existing multicast tree to accommodate the changes of membership.

Genetic algorithms (GAs for short) is a class of randomized algorithms for optimizing an objective function by breeding a population of possible solutions through time. In nature, we can observe "problems" being solved through evolution; species evolve as they adapt to dynamic environments. We extend this observation into the world of computer software via Evolver, the most powerful optimization package available. Evolver uses innovative GA technology to create environments where possible solutions continuously crossbreed, mutate, and compete with one another, until they "evolve" into the best solution. As a result, Evolver can find optimal solutions to virtually any type of problem, from the simple to the most complex.

A number of delay-constrained multicast routing algorithms have been proposed in the past few years, but they are all designed for static multicast groups. Though E.Biersack and J.Nonnenmacher [8] have presented a dynamic algorithm called WAVE for the multi-constrained QoS routing problem. However, it has some drawback that WAVE does not consider a given delay constraint explicitly. Wang et. al.[3], and Cheng and Li [4] have proposed some dynamic multicast routing heuristics. Nevertheless, the underlying multicast routing algorithms were designed for better-effort delivery only.

In this paper, a shared-tree routing protocol based on distributed genetic algorithms (GAs) is presented. The heuristic genetic algorithm is adopted to solve the minimal multicast tree with delay-constrained least-cost multicast routing problem. With this algorithm, a dynamic multicast routing tree which has an optimal network cost under the delay bound constraint can be constructed in a real time manner.

The article is structured as follows: Section 2, presents the distributed genetic algorithms. Section 3, exposes the distributed routing algorithm based on Genetic Algorithm. Section 4 specifies the simulation and the results in section 5.

2. GENETIC ALGORITHMS

GAs, first introduced by John Holland in 1975, are general-purpose stochastic optimization algorithms that use the process of biological evolution to generate new generations from superior “strains” of previous generations. Genetic algorithms are inspired by Darwin's theory about evolution. Solution to a problem solved by genetic algorithms is evolved.

Algorithm is started with a set of solutions (represented by chromosomes) called population. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (offspring) are selected according to their fitness-the more suitable they are the more chances they have to reproduce.

* This paper is supported by Nature science foundation of Hunan province (No.01JJY3015)

The following Outline of the Basic Genetic Algorithm

2.1[Start] The first step in the execution of the GA involves generating the set of the GA involves generating the set of initial links (parent links). A link contains information pertaining to the variables of the optimization problem. Generate random population of n chromosomes (suitable solutions for the problem) .

2.2[Fitness] Evaluate the fitness f(x) of each chromosome x in the population.

2.3[New population]Create a new population by repeating following steps until the new population is complete.

2.3.1[Selection] The selection section of the program calculates the average of the fitness for the population. If the fitness of a member is less than the average, the member does not survive. If the fitness of the member is more than the average, the member survives and its genes may be used in order to create new members in the cross-over section. (the better fitness, the bigger chance to be selected).

2.3.2[Crossover] The crossover section of the program generates new offspring in order to replace the members which have less fitness value than the average fitness value of the population. With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.

2.3.3[Mutation] With a mutation probability mutate new offspring at each locus (position in chromosome). The mutation section of the program is designed to introduce some variation to the population by replacing some genes with randomly generated values.

2.3.4**[Evaluate]** The evaluate section of the program evaluates each member of the population according to its genotype. The fitness value, *X*, is calculated using a specified formula. For example, we want to find the values for *A, B,* and *C* which give the maximum value of *X* for the following equation:

$$X = 100000/(A * A + 1) - (A + 1) * B + C$$

X is calculated and stored as the fitness of the member. Where *A* is stored as gene number one, *B* is as gene number two, and *C* is as gene number three.

2.3.5[Accepting] Place new offspring in a new population.

2.4[Replace] Use new generated population for a further run of algorithm.

2.5[Test] If the end condition is satisfied, stop, and return the best solution in current population.
2.6 [Loop] Go to step 2.2

3. DISTRIBUTED ROUTING ALGORITHM BASED ON GENETIC ALGORITHM

3.1 The distributed routing

Multicast is a mechanism used in communication networks that allows distribution of information from a single source to multiple destinations. The constitute of the distributed routing with six destination nodes is illustrated in fig.1.
A network can be modeled as a graph *N(V,E),* where *V* is a set of all nodes, representing routers or switches[2], *E* is a set of edges representing physical or logical connectivity between nodes. Each link is bi-directional. Let $s \rightarrow V$ be the multicast source, $M \subseteq V\text{-}\{s\}$ the multicast destinations, and $R+$ the set of positive real numbers. We define two additive functions to each link $e \rightarrow E$: the delay function delay $(e):E \rightarrow R+$ and the cost function $\text{cost}(e):E \rightarrow R+$. Let *t* be any destination node of *M*, and $p(s, t)$ be the path from s to t. For a given source node $s \rightarrow V$ and a destination node set *M*, there exist the following relationships for the multicast tree constructed by s and *M*.

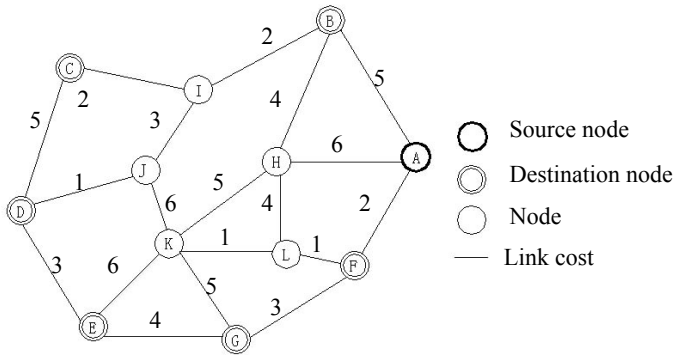


Fig.1 The constitute of distributed routing with six destination nodes

Definition 1:Delay-constrained least-cost multicast routing problem: Given a network $N(V,E)$, a source node $s \rightarrow V$, a destination node set $M \subseteq V\text{-}\{s\}$, the delay-constrained least-cost multicast routing problem is define as a multicast tree that satisfies:

$$\min\{cost(T(s,M)), T(s,M) \in T_c(s,M)$$

where $T(s,M)$ is the set of all delay-constrained multicast trees constituted by *s* and *M*. It has been demonstrated that the delay-constrained least-cost multicast routing problem is NP-complete.

Definition 2: The network is described as $G=(V,E,C,D)$, one of the multicast task $\prod = (S,T,c,d,w)$,where $S,T \subset V, |S| \geq 2, |T| \geq 2, c,d,w \in R^+$ .In order to find a multicast tree $\Phi^* + (V_\Phi, E_\Phi, C, D)$, it must satisfy the following conditions:

(1) $C(\Phi^*) \geqslant c.$
(2) $D(\Phi^*) \leqslant d.$
(3) $W(\Phi^*) \leqslant w$
(4) $\forall \Phi \in \Psi(\prod) : \text{Cost}(\Phi) \geq \text{Cost}(\Phi^*)$

3.2 The solving QoS multicast routing problem based on GA

The solving QoS multicast routing problem based on GA is shown in Fig.2.

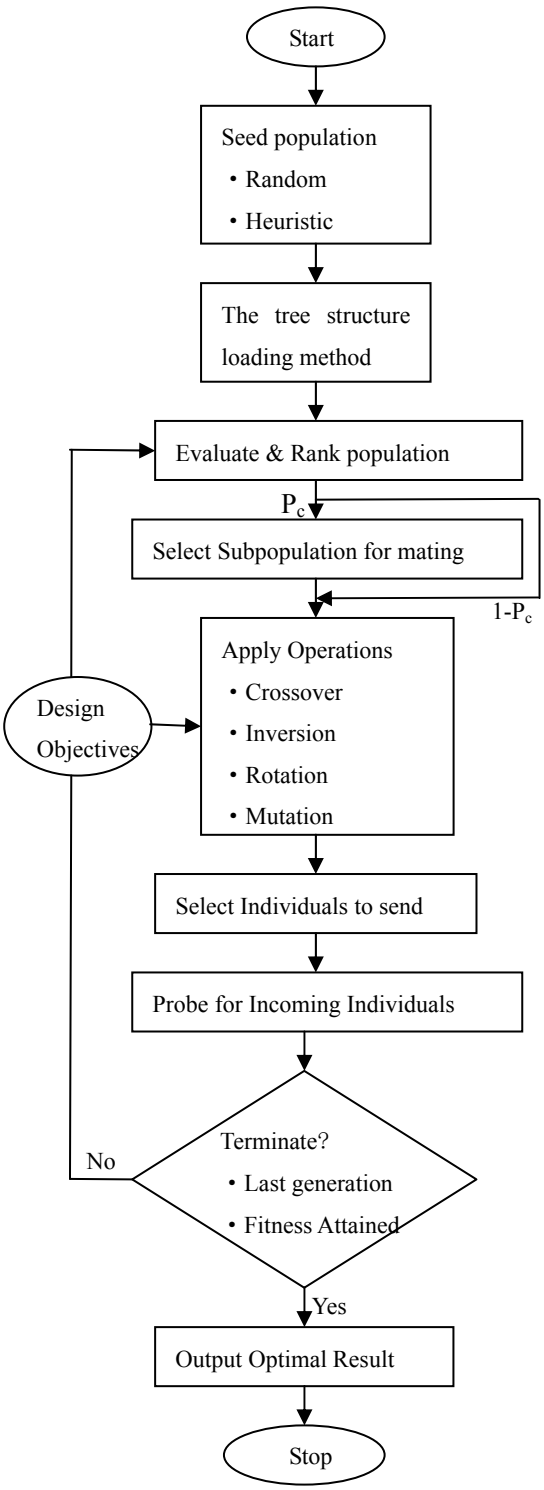


Fig.2 the flow diagram of the solving routing based on GA

4. SIMULATIONS

For the purpose of evaluating the efficiency of the proposed routing method. To compare the relative advantages or disadvantages of the GA algorithm with the other algorithms, a simulation study was conducted. The number of nodes varying in[20,200]. We have done the following experiments: (1) routing request ratio,(2) operation time testing , and (3) Variation in the mean delay as a function of the traffic level.

4.1 The routing request success ratio

The routing request success ratio θ_{req} is defined as follows:

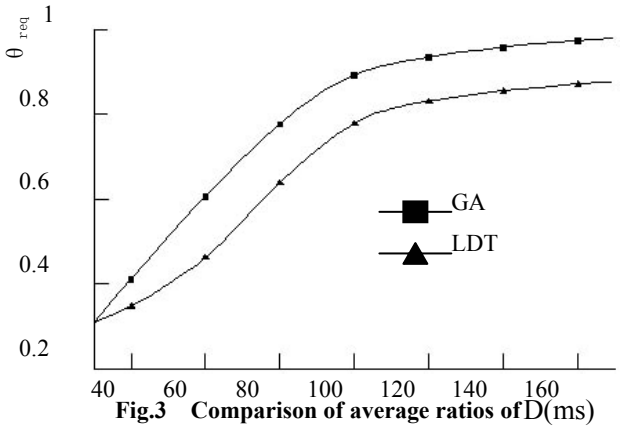


Fig.3 Comparison of average ratios of GA and LDT(N_{req}=100)

$$\theta_{req} = N_{ack} / N_{req}$$

where N_{ack} is the number of multicast routing requests accepted. N_{req} is the total number of multicast routing requests. We make a comparison to the success probability of the average request between GA and least multicast tree (LDT) algorithm, N_{req} =100, since the (LDT) has the highest routing request ratio of all multicast routing algorithms. From Fig.3, the success ratios of GA is always higher than that of LDT.

4.2 Operation time testing

With the number of nodes varying in [20,200], the operation time of GA algorithm is not much added. This proves the results of the operation time testing of GA algorithm as shown in Table 1.

4.3 Mean delay comparison

Fig.4. shows the variation of the mean delay as a function of the traffic level. The mean delays obtained by the GA method are in general more stable than those resulting from the other two methods.

5. CONCLUSIONS

GAs are general-purpose stochastic optimization algorithms that use the process of biological evolution to generate new generations from superior "strains" of previous generations. In this paper,. We present a Genetic algorithm for distributed multicast routing, including building and dynamic maintenance of multicast routing tree in package exchange network. The algorithm has the following characteristics:
the preprocessing mechanism, the tree structure coding method, the heuristic crossover technique, and the instructional mutation process. We have compared the performance of our algorithm with existing approaches and have demonstrated improved performance. The analysis of the algorithm presented, backed up the simulation results. The success ratios of GA is always higher than the exist optimal multicast routing algorithm. With the adding of the number of nodes, the operation times of GA is not much added. The mean delay obtained by the GA method are more stable than the other optimal routing algorithm.
With this algorithm, a dynamic constructed multicast routing tree which has a near optimal network cost under the delay bound constraint can be constructed in a real time manner.

Simulations shown that this algorithm is a simple, efficient, and scalable to a large network sizes under distributed multicast routing algorithm.

Table 1 Operation time of GA algorithm

Nodes	Routers	CPU (time)	Variations
20	32	0.26	0
40	95	0.42	1
80	180	1.22	0
100	188	2.32	1
120	230	3.82	0
140	288	5.86	2
160	328	6.81	1
180	370	9.21	3
200	429	9.46	1

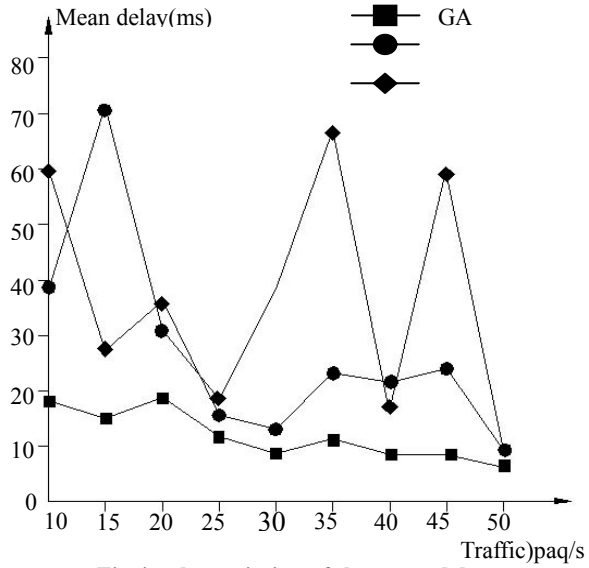


Fig.4 the variation of the mean delay

6. REFERENCES

[1] Wang Z,Crowcroft J.“Quality of service for supporting multimedia applications”, IEEE Journal on Selected Areas in Communications, 1996, 14(7):pp.1228-1234

[2] Wang Z,Shi Bing-xin,Liu Wei.“A distributed dynamic-constrained least-cost multicast routing heuristic”, Journal of software, China, 2001, 12(1),pp.1-10

[3] Chen M., Li Z.“A real-time multicast routing algorithm based on genetic algorithms”, Journal of solfware,China, 2001,12(5),pp.729-734

[4] Wang Z,Shi Bing-xin. “Solving QoS multicast routing problem based on he Kompella,V.P.,Pasquale,J, C.,Polyzos, G.C,Multicast routing for multimedia uristic genetic algorithm”, Chinese Journal of Computer. 2001,pp.55-61

[5] Waxman B.M. “Routing of multipoint connections”,IEEE TSAC,1988,6(9): pp. 1617-1622.

[6] Doar M.,Leslie I.,”How bad is native multicast routing”,IEEE INFOCOM’93. SanFrancisco, IEEE Press,1993.82~89. http://www.ieee-inform.org/1993

[7] .Parsa M., Zhu Q., Garcia-Luna-Aceves J.J. “An iterative algorithm for delay-constrained minimum-cost multicasting”, IEEE/ACM Transactions on Networking, 1998,6(4), pp. 461-474.

[8] Teck Sang Loha, Satish T.S.B ukkapatnamb.*, Deborah Medeirosa, Hongkyu Kwonb . A genetic algorithm for sequential part assignment for PCB assembly. Computer Industrial Engineering 40(2001) 293~307

[9] http://cs.felk.cvut.cz/~xobitko/ga/ Roman Novak,Joze rugelj,Gorazad Kandus. A note on distributed multicast routing in point-to point networks. Computers && operations research.PP1149~1156.

A Fault Tolerant Algorithm Based on Dynamic and Active Load Balancing for Redundant Services

Junwei Zhang, Junfeng Tian, Fengxian Wang
Mathematics and Computer College, Hebei University
Baoding, Hebei Province, 071002, P.R.China
E-mail: hebzhw@eyou.com

ABSTRACT

Under the circumstance of guaranteeing the availability of redundant service systems, a new fault tolerant algorithm SRAW (Some-Read-Any-Write) is presented in this paper for improving the performance. SRAW is based upon dynamic and active load balancing. By way of introducing the active load balancing into redundant service systems, it not only improves processing speed of requests but also achieves load balancing in a simple but efficient way. Integrated with consistency protocol in this paper, SRAW can also be applied to stateful services. The performance of the SRAW is also analyzed, and the comparisons with other fault tolerant algorithms, especially with RAWA, indicate that SRAW has efficiently improved the performance of redundant services without damaging system availability.

Keywords:redundant services, fault tolerant, active, load balancing, quorum

1. INTRODUCTION

The fault tolerant algorithms of present days are classified into two types: primary backup and active replication[1], but both of which have some limits, as discussed in [2]. ROWA (Read-Once-Write-All) algorithm tentatively balances the two types and classifies the requests into reads and writes. It provides a protocol similar to primary backup for 'reads' and an active replication for 'writes'. Hence ROWA does not reach real and efficient tradeoff. Although RAWA[2] (Read-Any-Write-All) balances them more efficiently, it also has some flaws. RAWA is a fault tolerant algorithm based on quorum[3], which is the set of redundant services that serve for a request, and RAWA can dynamically change the quorum between 1 and *N* (the number of all redundant services) with variability of system load. However it still bears some flaws as follows: (1) Requests of the same kind of service are carried out in serial way, which will reduce the system performance in distributed environment. (2) Too large communication expenses will aggravate the system's load.

Active load balancing[4](the receiver-initiated load balancing), achieving its operation in accordance with the load information of part of the system, has good scalability. It can be efficiently used in distributed systems. Its main idea is that the light load nodes don't passively receive but actively ask for the task from manager. Active load balancing has some merits: (1) It doesn't need exchanges of load information between nodes; (2) It lessens the load of the task manager and prevents the manager from becoming the system bottleneck, since most of the balancing operation is achieved by light load nodes.

Thus, based on active load balancing, a dynamic and active fault tolerant algorithm SRAW (Some-Read-Any-Write) is presented in this paper for improving the performance of redundant service systems. This design of SRAW algorithm bases on the following ideas: (1) By defining the request's quorum between 1 and N, SRAW algorithm reaches tradeoff between the primary backup and active replication; (2) The request's minimum quorum is decided by quantifying user's desire for availability, and request's quorum can be dynamically changed with the variability of the system load, therefore the user's desire for availability can be satisfied and the request's average processing speed can be increased; (3) By introducing active style, load balancing is simply but efficiently achieved and the task manager's load is reduced, even the requests can be carried out in parallel way and requests' average response speed can be increased; (4)A consistency protocol is presented for the stateful services. In this paper, the comparisons between SRAW and other fault tolerant algorithms, especially between SRAW and RAWA, show that the SRAW algorithm can efficiently improve the system performance and satisfy the distributed applications well.

2. ALGORITHM DESIGN

2.1. System model

SRAW algorithm bases on the following model. The environment is Ethernet with many hosts, which can communicate with each other. The communication is in asynchronous way that messages will not be lost, not need to be resent, it is in serial way without any destroyment; The redundant services are classified into stateful services and stateless services[5]. Stateful services, which capsulate both data and computation resources, can provide user with data and computation services, e.g. database server is a kind of stateful services. Stateless services, which only capsulate the computation resource, can only provide user with computation services, e.g. RPC can be looked on as a kind of stateless services; Faults in system are classified into fail-stop and Byzantine types. In this paper only fail-stop type is discussed;

In object-oriented viewpoint, services in system are abstracted as service classes which are achieved by many redundant services. The realization of redundant service is defined as instance, and instances of same service class can realize in different way. For example, the platform or program language may be different, but the instances must inherit the same service interface, we assume that instances of the same class stay on different hosts, which means that not more than one instance of same service class can stay on one host.

2.2. Algorithm frame and data structure

SRAW algorithm uses AMA (Administrator-Manager-Agent) frame, which consists of five parts: Administrator, OM(Object-Manager), HM(Host-Manager), SA(Server-Agent), RA(Request-Agent) , as illustrated in Fig.1.

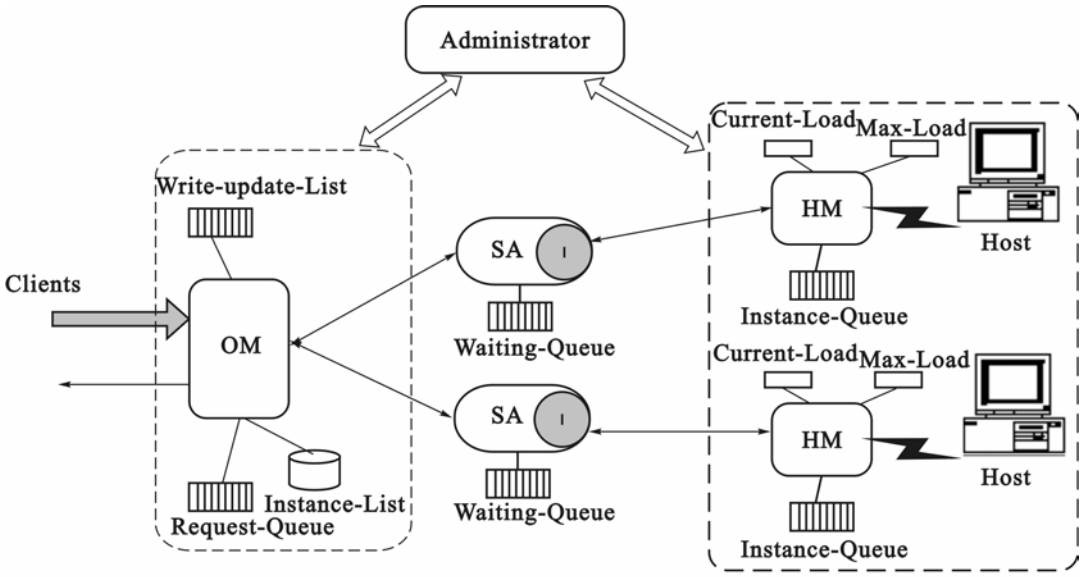


Fig.1 System Architecture of algorithm SRAW

Administrator, which has the information of all service objects and hosts, monitors the running condition of hosts and service instances by inspecting OMs and HMs periodically. Administrator can report the information to sys operator in real time, and can find the disabled OM and HM then retrieve them in time. OMs and HMs can therefore be assumed to be robust.

HM, which is host manager, stays on each host and manages information of the host, including host's Max-Load, host's Current-Load, and information of instances that stay on the host, and alternately including the service class of instance, Instance-Current-Load, etc. We assume that the number of instances that stay on the host H_i is N_i, the Waiting-Queue length of instance j is $L_j(1 \leqslant j \leqslant N)$, hence the host load index is defined as: Current-Load$_i = \sum_{j=1}^{N_i} L_j$. Host is weight loaded when Current-Load $\geqslant$ Max-Load, and light when Current-Load$<$Max-Load; The instance load index is defined as: Instance-Current-Load$_j$ = $L_j \times c(c \geqslant 1$, c is ratio) + Current-Load $- L_j$, that is Instance-Current-Load$_j$ = Current-Load + $L_j \times (c - 1)$. Instance is weight when Instance-Current-Load $\geqslant$ Max-Load, and light when Instance-Current-Load$<$Max-Load. The comparison between the definitions of host load index and instance load index shows that instance may be weight when host is light. Therefore, too long Waiting-Queue length of an instance is avoided.

OM manages the service information. It receives user's requests and transmits them to appropriated instances by load balancing. There are a Request-Queue, an Instance-List and a Write-Update-List in OM. Request-Queue saves the user's requests, including method name, parameters, result-type and request-type. Instance-list saves the information of the redundant services, including instance address, status and especially next-request-address according to which OM assigns proper request to the instance. Write-Update-List saves write-update-messages, including update address and update value. Besides, OM also manages the disabled redundant services and can retrieve and create them with information of other instances.

SA, the service proxy, which is the interface added upon instance realization, provides instance with management interface. SA decides whether or not to ask for task from OM by checking load status from HM. There is a Waiting-Queue which saves the requests that cannot be carried out in time. In Fig.1, I is the real service part and is transparent to users.

RA, which is user requests' proxy, stays on the client and receives the result from OM.

SRAW algorithm consists of two parts: dynamic and active fault tolerance and consistency protocol. The dynamic and active fault tolerance is only suitable to stateless services. For stateful services a consistency protocol is presented in this paper.

2.3. Dynamic and active fault tolerance

The main idea of dynamic and active fault tolerance is that OM dynamically changes the quorum of user requests in Request-Queue with the variability of the OM load, and then waits for redundant services not more than quorum actively asking for the request. Redundant service stays on a host and the host determines its load. When light loaded, redundant services actively ask for request from OM. Thus load balancing can be simply achieved and the OM's management load can be lessened, the OM's bottleneck can be avoided and the load information can be simply but efficiently collected in time.

Message types in system: Eight message types are defined in order to achieve the SRAW algorithm.

1) request-message(RequestSeq, MethodName, Parameters, Resulttype): request message sent from user to OM or from OM to SA.

2) null-task():message sent from OM to SA to mean "no new task".

3) free-report(InstanceAddr): asking for new request message sent from SA to OM.

4) inquiry-message(InstanceAddr): load inquiry message sent from SA to HM. InstanceAddr is the address of SA.

5) load-message(InstanceLoadStatus): load inquiry result message sent from HM to SA. InstanceLoadStatus is the sign of weight or light loaded.

6) executed-message():request executed message sent from SA to HM.

7) result-message(ResultSeq, Result): request result message sent from SA to OM or OM to user.

8) received-message(RequestSeq): a message sent from OM to SA to mean having received result.

Determination of the user request quorum: Minimum quorum of a request is determined by quantifying the user's desire for availability. When a new request arrives to the OM's Request-Queue, it aggravates the OM's load. Therefore we can set aside some system resources for the new request by dynamically reducing the quorum of previous requests to carry out the new request together with previous ones in parallel way. Every time when a new request arrives, the quorum can be set to be *N* since it is the latest, thus it can be carried out by all the redundant services to get more performance.

Algorithm step 1: The requests are input into the Request-Queue with FIFO order, and the quorum of the requests in Request-Queue is adjusted by OM. The Next-Request-Addr of SA in Instance-List, whose value is null, should be updated with the new request's address. Then OM waits for SA's free-report message.

Algorithm step 2: When the host is light loaded, SA sends the inquiry-message to HM periodically, if the InstanceLoadStatus in returned load-message is light, SA actively sends free-report message to OM, vice, versa.

Algorithm step 3: After receiving free-report message from SA, by inquiring the Next-Request-Addr of this SA in Instance-List, OM gets the request address in Request-Queue, which the SA asks for. Then OM checks the request: ①. If the request is null then OM sends the null-task message to SA and updates the Next-Request-Addr of SA in Instance-List with null. ②. If the request's quorum>0, OM assigns the request to SA and reduces its quorum by 1, at the same time OM updates the Next-Request-Addr with the next request's address in Request-Queue. ③.If the request's quorum=0, OM checks the next request in Request-Queue.

Algorithm step 4: After receiving the result-message from SA, OM searches for corresponding request in Request-Queue. At finding, OM sends result-message to user and delete the request in Request-Queue. If it fails, OM simply discards the result-message from SA.

2.4. Consistency protocol

For stateless services, the above algorithm selects a set of light load instances to carry out user's request, and thus assures the system with high availability and performance. But for stateful services, the above algorithm should be adjusted to keep the consistency of the redundant services: (1) The requests are classified into read and write types, both of which are treated accordingly; (2) A Write-Update-List is added in OM to improve performance for the write request.

The read request will not change the redundant service state, so the above algorithm can still be used. For write request, the quorum shall be set as *N* and can not be dynamically changed to make sure that every redundant service can meet once, and once only the write request.

After executing the write request, SA sends the result-message together with the changes of states to OM. After OM receives the result-message, OM creates write-update-message according to the result-message and puts it into Write-Update-List with FIFO order. At the same time OM marks the write request with executed sign. When other SA asks the write request again, OM sends the corresponding write-update-message to it. SA needs only to update the state without executing the request again. Hence the performance of write request can be improved greatly.

3. PERFORMANCE ANALYSIS

As discussed in [2], comparisons between active replication, primary backup, ROWA, RAWA algorithms show that the RAWA has best performance among them. Therefore we especially compare SRAW with RAWA in this paper.

3.1 Average response time

We assume that the number of all redundant services is *N*, the Request-Queue length is *L* and the request arrival rate is λ in both algorithms. In RAWA the serve time is t_{sr}, hence the response time t_r is:

$$t_r = t_{sr} + L/\lambda \tag{1}$$

In SRAW, the period of SA's asking for request is *T*, and request average quorum is *n*, thus the average parallel degree is *N/n* and the actual Request-Queue length is $L/(N/n)$, hence the response time t_s is:

$$t_s = t_{sr} + Tn/(2N) + nL/(N\lambda) \tag{2}$$

The comparison between formulas (1) and (2) shows that: $t_s \leqslant t_r$, when $T \leqslant \frac{2L}{\lambda}(\frac{N-n}{n})$.

3.2 Communication expense

We analyze the amount of messages that need to be sent while achieving one request in normal conditions. We assume that the number of redundant services is *N* and the write request arrival probability is ρ.

In RAWA, the messages include the broadcasts from OM to all redundant services, result-message from SA to OM and update messages from OM to the other *N*-1 redundant services. The message amount n_w is:

$$n_w = N + 1 + (N-1) = 2N \tag{3}$$

In SRAW, the free-report message, which SA sends periodically to OM, is very simple and can be integrated with heart-beaten, consequently the free-report message can be ignored. Thus the messages to achieve a read request include request-message from OM to SA and result-message from SA to OM, that is quorum+quorum. The messages to achieve a write request include one request-message from OM to SA, one result-message from SA to OM and *N*-1 write-update-messages, that is 1+1+*N*-1=*N*+1. Hence the message amount n_s is:

$$n_s = 2(1-\rho)quorum + \rho(N+1) \tag{4}$$

The comparison between formulas (3) and (4) shows that: $n_s \leq n_w$ in any case.

4. REALIZATION AND PERFORMANCE TESTING

In this paper, in an Ethernet with 5 nodes each of which runs Linux. We realize the system based on CORBA platform. Administrator is realized as the CORBA system service; OM and HM are realized as the CORBA objects. OM can stay on any node and HM stays on each node. HM is created by Administrator when host joins the system; RA and SA are auto created by the IDL compiler. RA is realized in client stub and SA is realized in server skeleton. Based on CORBA, the performances of RAWA and SRAW are tested contrastively.

In Fig.2, the curve shows the changes of response time in SRAW along with the period *T*'s changes in condition that the redundant service number is 3 and the request arrival rate is 200/sec. We can reach from the Fig.2 that the response time shortens along with *T*'s lessening and becomes less than that in RAWA when *T* is less than a certain value.

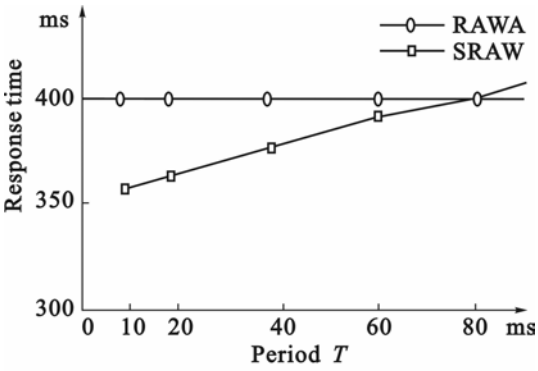


Fig.2 Variable curve of response time with different periods

In Fig.3, the curve shows the changes of response time along with the redundant services' number changes in condition that the *T* in SRAW is 40ms and the request arrival rate is 200/sec. We can reach from the Fig.3 that the response time shortens along with the redundant services' number increases in both algorithms, but the response time in SRAW shortens much more than that in RAWA, and even becomes less than that in RAWA when redundant services' number is more than a certain value.

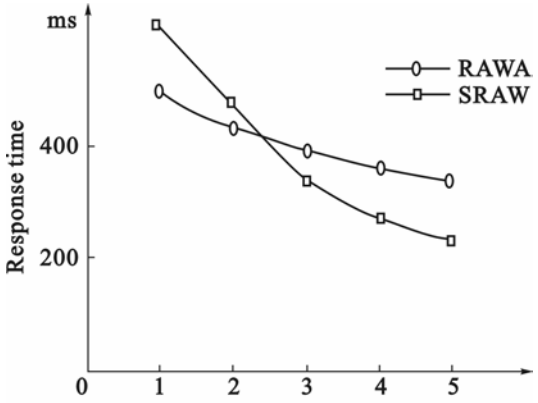


Fig.3 Variable curve of response time with different numbers of redundant services

5. CONCLUSION

This paper presents a fault tolerant algorithm SRAW, which, based on active mode, realizes a simple but efficient load balancing and improves the redundant service system performance. SRAW algorithm reaches efficient tradeoff between the active replication and primary backup by changing the quorum of user requests with the variability of the system loads. When the system is light-loaded, SRAW is similar to active replication, which makes the best of the system resources to carry out the user's request, thus not only the response speed can be increased but also the precious resources will not be wasted; When the system is weight-loaded, SRAW is similar to primary backup, which carries out the user's request with less redundant services in condition that the user's desire can be satisfied, thus the system resources can be set aside for other requests, which shows that SRAW bears predictability. Furthermore the redundant services consistency and the requests' executing in a parallel way are realized in simple and efficient way. Thus, conclusion can be drawn that SRAW improves the system performance efficiently.

6. REFERENCES

[1] Rachid,G.,Andre,S., "Softwre-Based replication for fault tolerance", IEEE Computer, 1997,30(4): 68~74

[2] Qian Fang, Jia Yan, Huang Jie, Gu Xiao-bo, Zou Peng. "A Dynamic Fault Tolerant Algorithm for Improving Performance of Redundant Services". Journal of Software, 2001,12(6): 928～935

[3] Mustaque,A.,Mostafa,H.A. "performance characterization of quorum-consensus algorithms for replicated data", IEEE Transaction on Software Engineering, 1989,15(4): 492~496

[4] Chen Hua-ping, Ji Yong-chang, Chen Guo-liang. "A Universal Model of Distributed Dynamic Load Balancing", Journal of Software, 1998,9(1): 25～29 (in Chinese)

[5] Silvano,M. "Client/Server term definition", In: Hemmendinger, D., Reilly,E.D.,eds. Encyclopaedia of Computer Science. Zurich: International Thomson Com.puter Publisher, 1998

A Time Synchronization Algorithm in Distributed Control Systems*

He Peng
Information Technology Center, Three Gorges University
Yichang, Hubei 443002, China
E-mail: hpeng@mail.ctgu.edu.cn
Li Jing
Information Technology Center, Three Gorges University
Yichang, Hubei 443002, China
E-mail: sx_kate@mail.ctgu.edu.cn
Xia Changhao
Department of Computer Science and Technology, Three Gorges University
Yichang, Hubei 443002, China
E-mail: xiachh@mail.ctgu.edu.cn

ABSTRACT

The application of time synchronization recently becomes common in many projects in China. This paper discusses a synchronization algorithm for distributed control systems. A model for the passive algorithm is introduced and a corresponding program of realization is included. The precision of time synchronization in the experiments is within the limits of microseconds.

Keywords: Distributed control system, Time synchronization, Synchronization algorithm, Winsock.

1. PREFACE

With the development of network technique, the distributed control systems are used more and more widely. Because the distributed control systems have no centralized unified time base, make the synchronous sampling of data and checking and analyzing of trouble losing the reference. Time synchronization is the key to eliminate difference between two kinds of timing system and the foundation of all applications in the distributed control systems. At present, there are many research achievements about multi-media message synchronization based on network transfer, but synchronous research of benchmark time in the distributed control systems is rarely.

This paper mainly discussed clock synchronization between host nodes in distributed control systems. Therefore, we must choose one of the host nodes as an authentic time source, and other host nodes only need to carry on time synchronization with it. This authentic time source is a common system time clock that called Time Server [1]. Therefore, in order to realize the time synchronization of distributed control system based on Local Area Network topology, this paper firstly analyzed the timing structure of single mainframe computer, then chose a passive algorithm to establish model and listed programs with Winsock.

2. TIMING STRUCTURE OF SINGLE HOST

There are three methods that realize the time synchronization of timing system: method of the hardware, software and statistic. Hardware method is mainly to realize higher precision of time synchronization by special timing devices, which directly use external base time frequency signal, such as high-stability frequency standard, 1pps pulse in GPS, etc., in order to get to absolute synchronization. Software method is generally to realize time synchronization by means of software command, system call and process message, mostly by the time stamp in multi-media message synchronization, which is suitable for relative synchronization. Statistic method is suited to the distributed system, which has higher precision by establishing suitable statistic model with mass synchronization message between nodes, to eliminate indetermination of time error. At some applications, such as distributed real-time control system, two or three methods can be used together.

However, any synchronization method is relative to the timing structure of computer. Fig1 is a three-level timing structure, which includes all single host system: RTC timing, BIOS timing, and OS timing. RTC timing is a peripheral equipment system that is relatively independent to host and only gives host the initial time when it starts. After getting the initial from RTC, BIOS begins to count at an inherent frequency independently of RTC. The counting seconds are stored in memory for OS to call, and the hour minute second style that OS counts by is relative to the counting seconds of BIOS.

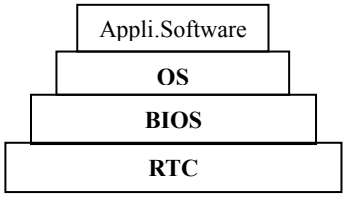


Fig.1 Timing structure of single host

Evidently, for single host system, there are three synchronization styles: synchronous RTC timing, synchronous BIOS timing, and synchronous OS timing. The former two are suitable for hardware method; the latter one is suit for software method, which is all the foundation to realize time synchronization of distributed control system. The approach discussed in this paper is based on reference [2], and according to the timing benchmark of Time Server, to realize synchronization of OS timing of other host nodes by serial interfaces or network interfaces.

3. ALGORITHM

Time synchronization of distributed control system is to realize the time synchronization of distributed control system

* This work is supported by the scientific and technological project of education department of Hubei Province. Grant No.2001B53005

based on LAN topology, therefore, according to its process features of realization, it includes centralized algorithm and distributed algorithm. Centralized algorithm refers that one of host nodes in distributed system is chosen as a Time Server which either has time base with high-stability to feed the need of time precision without external time base (called passive synchronization) or can supply approaches to synchronize with external time base (such as GPS) in order to get to higher precision (called active synchronization). No matter passive or active one, centralized algorithm needs the nodes of other host computers to realize time unified though the execution of algorithm, it includes passive and active algorithm.

The principal of passive algorithm is that every computer sends inquiring message to Time Server periodically at a period within $\delta/2P$. After receiving message, Time Server answers with real time value. In this case the Time Server is passive, so it is called passive algorithm and it is also called Cristian algorithm by reference [3].

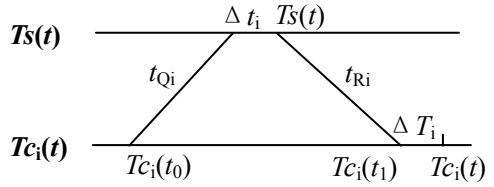


Fig.2 Synchronization time series asking for service

As shown in Fig.2, node *i* sends inquiring information at time $Tc_i(t_0)$, after t_{Qi} it gets to Time Server. The responding time Δt_i means after Δt_i Time Server sends time message. After transmission delay t_{R_i}, node *i* gains the correcting time $Tc_i(t_1)$, therefore:

$$Tc_i(t_1) = Ts(t) + \Delta t_i + t_{Ri} \quad (1)$$

For easy to measure, to node *i*, assume the round time is T_{Round},

$$T_{Round} = Tc_i(t_1) - Tc_i(t_0) = t_{Qi} + \Delta t_i + t_{Ri} \quad (2)$$

When round transmission time is equal i.e. $t_{Qi} = t_{Ri}$, therefore:

$$T_{Ci}(t_1) = Ts(t) + (T_{Round} + \Delta t_i)/2 \quad (3)$$

$$\Delta T_i = T_{Ci}(t) - T_{Ci}(t_1) \quad (4)$$

Because measurement and estimating T_{Round} and Δt_i have nothing to do with transmission route, we'd better calculate ΔT_i only on local computer for realization of synchronization.

4. WINSOCK REALIZATION BASED ON TCP/IP

TCP/IP is a protocol suite that is actually a connection-oriented protocol with functions of flow control, reorder, multiplex and failure tolerant etc. This paper chooses connection-oriented Client/Server computing mode to realize passive algorithm of time synchronization and use MFC developing tool of WinSock 2.0 to finish the program design. The host operation system is Win98 and Win NT that support WinSock 2.0. The Local Area Network protocol is Ethernet protocol based on bus topology, and the data compact technology is used in Winsock transmission [4].

MFC supplies two kinds of class to handle network communication of Windows Sockets in which CAsyncSocket class has described a principal programming interface of Windows Sockets, and CSocket class derived from CAsyncSocket class is a kind of abstract to WinSock at a higher level. CAsyncSocket class has enveloped Windows Sockets and has supplied object-oriented software package on Winsock API as an end of network communication link. CSocket class has inherited a lot of member functions of CAsyncSockets class and provided Winsock API with higher level in which there is an important feature that has a function called automated jam handling.

In order to realize the passive algorithm based on Client/Server computing mode using Socket class, firstly a listening Socket in Time Server should be established to monitor the connect require from host node No. *i*, secondly a Socket requiring connect in host node No. *i* at client aspect is established, after Time Server being accepted the require message, the Socket corresponding to that node has been created, and data link has been established. Therefore, Time Server and host node No.*i* at client aspect can make data interchange by this data link. The communication operation order to create byte stream in Client/Server computing mode is as following:

at Time Server

① ClistenSocket SockServer;
//Construct a Socket class
② SockServer.Create (nPort);
//Create WinSocket
③ SockServer.Listen();
//Begin to listen to connect require
④ CClientSocket SocketClient;
//Construct a new Socket class
⑤ SockServer Accept (SocketClient);
//Accept connect
⑥ SockServer.Send (Data, Length); or
Sockserver.Receive (Data, Length);
//Start data communication

at Host Node *i*

① CClientSocket SockClient;
//Construct a Socket class
② SocketClient.Create();
//Create WinSocket
③ SocketClient.Connect(ServerIP,nPort);
//Ask for connection
④ SockClient.Receive (Data, Length); or
SockClient.Send (Data, Length);
//Start data communication

Above orders show that Timer Server has two CSocket subclasses CListenSocket and CclientSocket. The former is used for the class listening to link require, the latter is for Creating WinSocket corresponding to different client and CSocket class that make data communication. Host node No.i in client aspect only has one CClientSocket class. When link require arrives, Time Server creates the CClientSocket class corresponding to that client and then makes data communication using the member functions Receive (Data, Length) and Sends (Data, Length) of that class, where Data is byte stream pointer that point to real time message packed by data compact technology, Length is the length of packed real time message. As space is limited, only two sub-class definitions of Time Server are given and program design of every client host nodes is the same.

CListenSocket was defined as follows:

```
Class CListenSocket: public CSocket
  { public: //Attribute
    CShengView    *my_pv; //view class pointer
    public: //Operation
```

```
        ClisteningSocket (CSheng View *pv);
                                // Construct function
        virtual ClisteningSocket (); //Create function
        virtual void OnAccept (int nErrorCode);
                                //Virtual function
        }

   //Add client's process code in OnAccept(int nErrorCode)
   void CListeningSocket::OnAccept(int nErrorCode)
      { CSocket::OnAccept(nErrorCode);
                //Accept process code of require message
        my_pv->ProcessPendingAccept();
                //Call member functions of view class to
                  process FD_ACCEPT message further
        }
```

CClientSocket was defined as follows:

```
   Class CClientSocket: public CSocket
      { public: //Attribute
        CShengView    *my_pv; //View type pointer
        public: //Operation
        CClienterSocket(CShengView *pv);
                        //Construct function
        virtual CClienterSocket(); // Create function
        virtual void OnReceive(int nErrorCode);
                                //Virtual function
        virtual void OnClose(int nErrorCode);
                                //Virtual function
}

   //Add client's process code in OnReceive(int nErrorCode)
   void CClienterSocket::OnReceivet(int nErrorCode)
      { CSocket::OnReceive(nErrorCode);
                //Process code to receive data message
        my_pv->ProcessPendingRead(it);
                //Call member functions of view class to
                  process FD_READ message further
        }

   //Add client's process code in OnClose(int nErrorCode)
   void CClienterSocket::OnClose(int nErrorCode)
      { CSocket::OnClose(nErrorCode);
                //Close process code of connect message
        my_pv->CloseSocket(it);
                //Call member functions of view class to
                  process FD_CLOSE message further
        }
```

5. ANALYSE AND CONCLUSION

In this paper, time synchronization technique of distributed control system is discussed from point of view of application and program is realized by choosing a kind of passive algorithm. This algorithm must have a time-base of high stability for the Time Server or main node. Otherwise it must have a measure that can synchronize with external time source to realize time synchronization between client computer and Time Server by software programming. The algorithm is used in a supervisory control system of some power station in Gezhouba. The real-time statistic and analyzed data show that time synchronization precision is within 1ms[5]. Therefore, the algorithm is feasible and the program design is suitable. But it is worth noticing that before actually executing the algorithm, refreshing period(δ /2P) should be computed to decide refreshing frequency.

6. REFERENCES

[1] He Peng, Xia Changhao, Wu Haitao, “Study on Time Synchronization of Distributed System”, DCASBES2001 Proceedings, Science and Technology Press: 192-194

[2] Andrew, Tanenbaum Z., “Distributed Operation System”, Beijing: Electronic Industry Press, 1999.

[3] Cristian F., “Probabilistic Clock Synchronization”, Distributed Computing, 1989(3): 146-158

[4] He Peng, Wu Haitao, “A Data Compact Technology for Socket Transmission”, Journal of University of Hydraulic and Electric Engineering/Yichang, 1998(6): 33-36

[5] He Peng, Zeng Weilu, “Study on the Time Synchronization Technology Used in SCADA System”, Journal of North China Electric Power University, 2000(7): 47-49

On Convergence Bounds of GMRES Algorithm

Gang Xie
Institute of Computer Applications, CAEP, China.
E-mail: xieg@caep.ac.cn

ABSTRACT

In this paper we first make a brief review of GMRES convergence results. Then we derive new bounds for the GMRES residual norm by making use of a unitary matrix U and a Hermitian positive definite matrix P which are GMRES-equivalent to the coefficient matrix A with respect to the initial residual r_0. The existence of such U and P was proved by Leonid. As a GMRES residual norm bound for linear systems with Hermitian positive definite coefficient matrices is known and a GMRES residual norm bound for linear systems with unitary coefficient matrices can be readily derived from Liesen's work, our new bounds follow from the fact that two GMRES-equivalent matrices make the same residual.

Keywords: Systems of linear equations, iterative method , GMRES, GMRES-equivalent matrix, convergence bounds **.**

1. INTRODUCTION

The GMRES algorithm by Saad and Schultz is a popular iterative method for solving systems of linear equations:

$Ax = b$, $A \in C^{N\times N}$, $b \in C^N$. (1.1)

Given a initial guess x_0 for the solution of (1.1), GMRES yields iterates x_n so that

$$\|r_n\| := \|b - Ax_n\| = \min_{p_n \in \pi_n} \|p_n(A) r_0\| \qquad (1.2)$$

where $\|.\|$ denotes the Euclidean vector and corresponding matrix norm, $r_0 := b - Ax_0$ is the initial residual, and π_n denotes the set of polynomials of degree n with value 1 at the origin. A strict definition of the GMRES algorithm follows.

Definition 1.1. For $n = 1,2,...$, let

$$x_n \in x_0 + k_n(A, r_0)$$

be a vector , such that

$$r_n := b - Ax_n \in r_0 + Ak_n(A, r_0)$$

satisfies

$$r_n \perp Ak_n(A, r_0),$$

then the vectors x_n and r_n are called n th GMRES iterate and residual, respectively.

The convergence analysis of GMRES has been an active field of research in recent years and it is generally agreed that it needs further work. Using characterizations of matrices that generate the same GMRES residuals—so called GMRES-equivalent matrices—Liesen [1] derived bounds that depend on the initial guess. Leonid Knizhnerman [2] constructed a unitary matrix and a Hermitian positive definite matrix which are GMRES-equivalent to A. In this paper we first sum up various results on the convergence bounds of GMRES. Then based on the works of [1] and [2], we derive new bounds on GMRES residual norm by making use of the idea of GMRES-equivalent matrix. These new bounds imply an exponential convergence of the GMRES residual.

2. A REVIEW OF GMRES CONVERGENCE RESULTS

Let us briefly review the most important convergence results for GMRES.

2.1. Bound for diagonalizable matrix

Suppose A is diagonalizable, $X^{-1}AX = D, D$ is diagonal. Then the GMRES residual satisfy

$$\|r_n\|/\|r_0\| \le \kappa(X) \min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(A)} |p_n(\lambda)| . \qquad (2.1)$$

This bound was the first convergence result for GMRES. If A is normal, then $\kappa(X) = 1$ and (2.1) is sharp in the sense that for each n , there exists an initial guess for which equality holds. For diagonalizable but nonnormal A , $\kappa(X)$ might be very large, and the right-hand side of (2.1) might be a large overestimate of the actual residual norm. It was shown by Greenbaum that any nonincreasing curve of residual norms,

$$\|r_0\| \ge \|r_1\| \ge ... \ge \|r_{N-1}\| > \|r_N\| = 0$$

can be produced by GMRES applied to a normal, even unitary $N \times N$ matrix.

2.2. Bound based on the pseudospectra of the matrix

A bound on GMRES residual norm can also be derived using ε -pseudospectra of A , namely

$$\|r_n\|/\|r_0\| \le \frac{L_\varepsilon}{2\pi\varepsilon} \inf_{p_n \in \pi_n} \max_{\lambda \in \Lambda_\varepsilon(A)} |p_n(\lambda)| \qquad (2.2)$$

where L_ε denotes the arc length of $\partial\Lambda_\varepsilon(A)$[3]. Note that (2.2) holds for all linear systems, while (2.1) requires A to be diagonalizable. Generally, it is very costly to compute $\Lambda_\varepsilon(A)$, in particular when the cost of this computation is compared to the cost of a single GMRES run. Both (2.1) and (2.2) sometimes can not describe the rate of convergence of GMRES, because no bound on their right-hand side is known. We also note that it is possible to construct examples in which the right-hand side of (2.1) overestimate the left-hand side considerably.

2.3. Bounds based on the field of values of the matrix

Another bound on GMRES residual norm is based on the field of values $F(A)$. Defining $\nu(A) := \min_{z \in F(A)} |z|$, Starke[4] showed that

$$\|r_n\|/\|r_0\| \le \left(1 - \nu(A)\nu(A^{-1})\right)^{n/2} \qquad (2.3)$$

This bound gives a reduction factor for the residual norm, provided that $0 \notin F(A)$ and $0 \notin F(A^{-1})$, which excludes

indefinite matrices. It is possible to obtain approximations to the right-hand side of (2.3) during the GMRES run by solving small eigenproblems. Thus (2.3) is interesting from a practical point of view. There exists a second bound based on the field of values, namely

$$\|r_n\|/\|r_0\| \le 2c_n\left[R(F(A))\right]^n \quad , \tag{2.4}$$

where the c_n are some positive constants satisfying $c_n < 2/\left(1-\left[R(F(A))\right]^n\right)$. As in the case for (2.3), the bound (2.4) requires that $0 \notin F(A)$. In most cases it is expensive to compute $R(F(A))$. Also note that if $R(F(A)) \approx 1$, then the constant c_n might be very large[5]. With regard to this, we point out that in all examples of recent study by Ernst, computed asymptotic convergence factors $R(F(A))$ range between 0.995 and 0.9999, although GMRES shows very different types of convergence behavior.

2.4. Bounds depending on the initial guess

Note that none of the bounds presented above depends on r_0.

Since $\Lambda(A)$, $\Lambda_\varepsilon(A)$ and $F(A)$ are invariant under unitary similarity transformation, all of the above bounds are the same for the linear systems (1.1) with A replaced by $U^H AU$ for any unitary matrix U. But the practical behavior of GMRES applied to these systems with the same initial guess might be completely different. For example, the algorithm may terminate at step one for one system, while need many more steps to terminate for another, although all of the above bounds gave them the same convergence estimate.

Not considering the dependence of GMRES on the initial residual can also lead to a misprediction of the worst-case behavior. Since GMRES minimizes $\|p_n(A)r_0\|$ for a given r_0, "worst-case GMRES" is the maximization of $\min_{p_n \in \pi_n}\|p_n(A)r_0\|$ over all r_0 with norm one. On the other hand, all of the above bounds are derived from the "ideal GMRES" approximation problem

$$\min_{p_n \in \pi_n}\|p_n(A)\| .$$

It is clear that "ideal GMRES" forms an upper envelope for all possible actual GMRES convergence curves. However, there are examples for which this envelope is never attained and the ratio of left- to right-hand side can be made arbitrarily small, which means the overestimate is arbitrarily large.

Liesen[1] derived bounds that depend on the initial guess and are thus conceptually different from standard "worst-case" bounds. Furthermore, approximations to these bounds can be computed from information generated during the run of a certain GMRES implementation. In numerical experiments it was often observed that for small n, close approximations to these bounds could be obtained. The approximations then allow prediction of how the algorithm will perform in later stages of the iteration. The bounds look like

$$\frac{\|r_n\|}{\|r_0\|} \le 4\frac{\kappa(R)}{\gamma^n - 1}$$

and

$$\frac{\|r_n\|}{\|r_0\|} \le \frac{4}{\hat{\gamma}^n - 1}$$

where $\gamma := \left(\cos\frac{\phi}{4}\right)^{-1}$ and $\hat{\gamma} := \left(\cos\frac{\hat{\phi}}{4}\right)^{-1}$. For detailed information about how these bounds were derived see [1].

2.5. Bound for Hermitian positive definite matrix

Using translated Chebyshev polynomials we can derive from (2.1) the following bound

Theorem 2.5. If A is Hermitian positive definite and $\kappa := \kappa(A)$, then

$$\frac{\|r_n\|}{\|r_0\|} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n \qquad 1 \le n \le N \tag{2.5}$$

3. GMRES-EQUIVALENT MATRIX

In the following, by r_n^{M,r_0} we denote the nth residual, when GMRES is applied to a linear system with the matrix M and initial residual r_0.

Definition 3.1. Suppose that $A \in C^{N\times N}$, $r_0 \in C^N$ and $d := \dim AK_N(A, r_0)$. Then $B \in C^{N\times N}$ is called GMRES-equivalent to A with respect to r_0 if $\dim BK_N(B, r_0) = d$ and $AK_n(A, r_0) = BK_n(B, r_0)$ for $1 \le n \le d$.

Theorem 3.1. If B is GMRES-equivalent to A with respect to r_0, then

$$r_n^{A,r_0} = r_n^{B,r_0} .$$

The result of this theorem can be easily derived from Definition 1.1 and Definition 3.1. Using Theorem 3.1 we can relate the convergence of GMRES for the given linear system to the convergence for some other system, which is easier to analyze. Our starting point for the derivation of the new bounds therefore is the construction of suitable matrices that are GMRES-equivalent to A with respect to the given r_0.

4. NEW BOUNDS ON GMRES RESIDUAL NORM

In this section we make use of Liesen's[1] and Leonid's[2] results to derive new bounds for GMRES residual norm.

4.1. Bound derived from GMRES-equivalent unitary matrix

Definition 4.1. Suppose that the spectrum of the $d \times d$ unitary matrix U is given by $\Lambda(U) := \left\{e^{i\beta_j} : 0 \le \beta_1 \le \beta_2 \le \ldots \le \beta_d < 2\pi\right\}$ and let $\beta_{d+1} := 2\pi + \beta_1$. We define the d gaps in $\Lambda(U)$ by $g_j := \beta_{j+1} - \beta_j$ for $1 \le j \le d$.

$\phi = \max\{g_1, g_2, \ldots, g_d\}$ denotes the largest gap in $\Lambda(U)$.

Proposition 4.1 [1]. For every unitary matrix U and every

$\alpha \in R$, we have

$$\min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(U)} |p_n(\lambda)| = \min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(e^{i\alpha}U)} |p_n(\lambda)| .$$

Therefore, if we define

$$\Omega_\phi := \left\{ e^{i\alpha} : \frac{\phi}{2} \le \alpha \le 2\pi - \frac{\phi}{2} \right\},$$

then we get

$$\min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(U)} |p_n(\lambda)| \le \min_{p_n \in \pi_n} \max_{z \in \Omega_\phi} |p_n(z)| . \qquad (4.1)$$

By constructing a suitable polynomial $p_n(z)$ in (4.1), the following Lemma can be derived from this proposition.

Lemma 4.2 [1]. If U is a unitary matrix, ϕ is the largest gap in $\Lambda(U)$ and $\gamma = \left(\cos \frac{\phi}{4} \right)^{-1}$, then

$$\min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(U)} |p_n(\lambda)| \le \frac{4}{\gamma^n - 1} .$$

Lemma 4.3. If $r_{N-1}^{A,r_0} \neq 0$, then there is a unitary matrix U that is GMRES-equivalent to A with respect to the given r_0.

For a proof of Lemma 4.3 see Leonid [2]. Now we are ready to give our new bound for the GMRES residual.

Theorem 4.1. If $r_{N-1}^{A,r_0} \neq 0$, then there exists a constant γ (>1) such that

$$\frac{\left\| r_n^{A,r_0} \right\|}{\left\| r_0 \right\|} \le \frac{4}{\gamma^n - 1} \qquad 1 \le n \le N . \qquad (4.2)$$

Proof: According to Lemma 4.3, we have a unitary matrix U that is GMRES-equivalent to A with respect to r_0. So from Theorem 3.1 and Lemma 4.2 we get

$$\begin{aligned} \left\| r_n^{A,r_0} \right\| &= \left\| r_n^{U,r_0} \right\| \\ &= \min_{p_n \in \pi_n} \left\| p_n(U) r_0 \right\| \\ &\le \min_{p_n \in \pi_n} \max_{\lambda \in \Lambda(U)} |p_n(\lambda)| \\ &\le \frac{4}{\gamma^n - 1} . \end{aligned}$$

Lemma 4.4 [2,Theorem 6.3]. If for some constant $p(>1)$ the residual norms satisfy the inequality

$$\left\| r_k^{A,r_0} \right\| / \left\| r_{k+1}^{A,r_0} \right\| \ge p \qquad k = 0,1,\ldots,N-2 ,$$

then the unitary matrix mentioned in Lemma 4.3 satisfies

$$\left\| (U - I^{-1}) \right\| \le \frac{1}{2} \sqrt{\frac{p+1}{p-1}} \left(\frac{1}{\sqrt{1 - p^{-2}}} + \frac{1}{\sqrt{p^2 - 1}} \right) .$$

Using Lemma 4.4 we can derive a lower bound for the largest gap ϕ in $\Lambda(U)$. In fact, for any $e^{i\beta} \in \Lambda(U)$ there exists a vector x such that

$$Ux = e^{i\beta} x .$$

Hence

$$\begin{aligned} (U - I)x &= (e^{i\beta} - 1)x \\ x &= (e^{i\beta} - 1)(U - I)^{-1} x . \end{aligned}$$

So

$$\|x\| \le \left| e^{i\beta} - 1 \right| \left\| (U - I)^{-1} \right\| \|x\| ,$$

and so

$$\left| e^{i\beta} - 1 \right| \ge \left\| (U - I)^{-1} \right\|^{-1} .$$

This means there is a lower bound on ϕ and so the γ in (4.2) has a constant lower bound $\gamma_0(>1)$. Furthermore γ_0 can be expressed with the value p in Lemma 4.4.

4.2. Bound derived from GMRES-equivalent Hermitian positive definite matrix

Assume $r_{N-1}^{A,r_0} \neq 0$ and define the numbers

$$g_k = \sqrt{\left\| r_{k-1}^{A,r_0} \right\|^2 - \left\| r_k^{A,r_0} \right\|^2} \quad \text{if } 1 \le k \le N-1$$

$$g_k = \left\| r_{N-1}^{A,r_0} \right\| \quad \text{if } k = N$$

Lemma 4.5. If $g_i \neq 0$, $i = 1,2,\ldots,N$, then there exists a Hermitian positive definite matrix p which is GMRES-equivalent to A with respect to r_0. Furthermore, if g_k satisfies the inequality

$$g_i / g_j \le c p^{i-j} , \quad c > 0,\ 0 < p < 1,\ N \ge i \ge j \ge 1 ,$$

then the condition number of p is estimated by

$$\kappa(p) \le \left[\frac{c(1+cp)}{1-p} \right]^2 .$$

This Lemma can be easily derived from Leonid [2]. Now we are ready to derive another bound for the GMRES residual from Lemma 4.5, Theorem 3.1, and Theorem 2.5.

Theorem 4.5. If $r_{N-1}^{A,r_0} \neq 0$ and $g_k > 0$ $1 \le k \le N-1$, then there exists a constant κ such that

$$\frac{\left\| r_n^{A,r_0} \right\|}{\left\| r_0 \right\|} \le 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^n \qquad 1 \le n \le N .$$

Furthermore, if $g_i / g_j \le c p^{i-j}$ for $c > 0, 0 < p < 1, N \ge i \ge j \ge 1$, then κ can be estimated by

$$\kappa \le \frac{c(1+cp)}{1-p} .$$

5. CONCLUSION

By making use of a unitary matrix U and a Hermitian positive definite matrix p which are GMRES-equivalent to A with respect to r_0, we get new bounds for GMRES residual. As these bounds are determined by the largest gap in the spectrum of the unitary matrix and the condition number of the Hermitian positive definite matrix respectively, it is meaningful to further investigate the bounds on these gap and condition number.

6. REFERENCES

[1] LIESEN, Computable convergence bounds for GMRES, SIAM J. MATRIX ANAL APPL. Vol. 21,

No. 3, pp. 882-903, 2000.
[2] LEONID KNIZHNERMAN, On GMRES-equivalent bounded operators, SIAM J. MATRIX ANAL APPL. Vol. 22, No. 1, pp. 195-212, 2000.
[3] N. M. NACHTIGAL, S. REDDY, AND L. N. TREFETHEN, How fast are nonsymmetric matrix iterations? SIAM J. MATRIX ANAL APPL. 13(1992), pp.778-795.
[4] G. STARKE, Field-of-values analysis of preconditioned iterative methods for nonsymmetric elliptic problems, Numer. Math., 78(1997), pp.103-117.
[5] O. ERNST, Residual-minimizing Krylov subspace methods for stabilized discretizations of convection-diffusion equations, SIAM J. MATRIX ANAL APPL. Vol. 21, No. 4, pp. 1079-1101, 2000.

Parallel Chaotic WR Algorithms for Discretized Dynamic System

Dongjin Yuan
Department of Mathematics,
Yangzhou University, Yangzhou, JiangSu 225002, P.R.of China
E-mail: dongjinyuan@yahoo.com

ABSTRACT

In this paper we establish several algorithms of discretized parallel chaotic waveform relaxation methods for solving linear ordinary differential systems based on some given models. Under some different assumptions on the coefficient matrix A and its multisplittings we obtain corresponding sufficient conditions for convergence of the algorithms. Also a discussion on convergence speed comparison of synchronous and asynchronous algorithms is given.

AMS *subject classification:* 65L05 65F10
Key words: Multisplitting, parallel, chaotic, convergence, waveform relaxation, H-matrix.

1. INTRODUCTION

Let us consider linear *m*-dimensional systems of the form

$$x'(t) + Ax(t) = f(t), \quad x(0) = x_0, \tag{1}$$

with $t \in [0,T], x \in C^1([0,T];R^m), f \in C([0,T];R^m)$, $A \in R^{m\times m}$ .The decoupling process can mathematically be described by a splitting of the matrix

$$A = M - N. \tag{2}$$

The waveform relaxation algorithm is given by the iteration:

$$x'_{n+1}(t) + Mx_{n+1}(t) = Nx_n(t) + f(t), x_{n+1}(0) = x_0. \tag{3}$$

As a starting function for x_0 usually constant initial values are chosen, i.e., $x_0(t) \equiv x_0$ .Iteration (3) is solved not only for one timepoint but for a fixed time interval [0, T_0] which is usually a subset of the domain of integration [0, T]. After convergence has occurred on the window [0, T_0] one proceeds to the next window $[T_0, T_1]$ until finally the end of the domain of integration T is reached.

It is worthy of remark that in [8], the author used an implicit integration method by parallel multisplitting, which is synchronous, for solving (1) and obtained the convergence of the method, in which the matrix A in (1) is only an M-matrix.

The main purpose of this paper is to establish three parallel chaotic algorithms based on some given models in [2, 6, 10] for solving linear ordinary differential systems (1), in which the matrix A can be an H-matrix, and investigate the corresponding convergence of these algorithms.

2. NOTATION AND ALGORITHMS

A splitting (2) of a matrix A 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 integer $L \geq 2$ a multisplitting of $A \in R^{m\times m}$ is a collection of L triples (M_l, N_l, E_l) of $m \times m$ real matrices, $l = 1,2,\cdots,L,$, for which each E_l is nonnegative diagonal, each M_l is invertible and the equations

$$A = M_L - N_l, \qquad l = 1,2,\cdots,L \tag{4}$$

and

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

are satisfied.

We can now split the matrix A L times and we will take the right-hand side as input. $y_{l,n+1}(t)$ will denote the unknowns of the l th splitting. We then have

$$y'_{l,n+1}(t) + M_l y_{l,n+1}(t) = N_l x_n(t) + f(t), \tag{6}$$
$$l = 1,\cdots,L,$$
$$y_{l,n+1}(0) = x_0 .$$

In the case of synchronous communication, after having solved each subsystem and since there is no iteration between two different subsystems we compute a new approximation to the solution of (1) by

$$x_{n+1}(t) = \sum_{l=1}^{L} E_l y_{l,n+1}(t). \tag{7}$$

Since we are using equidistant timepoints we will denote by t_r the timepoint $t_0 + rh$. Let t_0 denote the starting point of the actual window which for simplicity will be assumed to be zero. Furthermore we will use $y_{l,n,r}$ to denote the approximation of the l th subsystem during the nth sweep at time t_r, $x_{n,r}$ the approximation of the solution of (1) at time t_r , f_r the evaluation of the function f at time t_r , and $x(t_r)$ the exact solution of (1) at time t_r , respectively.

If we use the implicit Euler method for solving the l th subsystem numerically, we get the following relation:

$$y_{l,n+1,r} = (I + hM_l)^{-1}(y_{l,n+1,r-1} + h(N_l x_{n,r} + f_r)) . \tag{8}$$

We denote

$$B_l = (I + hM_l)^{-1}, \quad l = 1,2,\cdots,L.$$

Since we are studying the convergence over the whole window, we are only interested in the dependence of the $(n+1)$st approximation on the nth approximation. Using (8), the (n+1)st approximation $x_{n+1,r}$ of $x(t_r)$ at timepoint r is given by:

$$x_{n+1,r} = \sum_{l=1}^{L} E_l y_{l,n+1,r} \tag{9}$$

where

$$y_{l,n+1,r} = F_l(x_{n,r})$$

and

$$F_l(x_{n,r}) = hB_l N_l x_{n,r} + hB_l f_r + B_l y_{l,n+1,r-1}. \tag{10}$$

For a nonnegative integer $\mu_{l,n}$, in which l, the number of the processor and i, the index of the iteration step, let

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

The $F_l^{\mu_{l,n}}$ is the $\mu_{l,n}$ -th composition of the affine operator.

Using the above notation, (8),(9) and some given models in [2,6,10] we can now describe three algorithms based on the discretized parallel chaotic waveform relaxation method.

Algorithm 2.1 *Suppose that* $x_{0,r}$ *is the initial approximation of* $x(t_r)$ *at timepoint* r *and* $F_l(x_{n,r})$ *is satisfying (10). For* $n = 1,2,\cdots,$ *until convergence, perform*

$$x_{n+1,r} = \sum_{l=1}^{L} E_l F_l^{\mu_{l,n}}(x_{n,r})$$

with $\mu_{l,n} \geq 1, \quad rh \leq T,$

where $\mu_{l,n} = \lfloor \max_{1\leq l\leq L}(\tau_{l,n}) / \tau_{l,n} \rfloor, \quad l = 1,2,\cdots,L,$ with $\tau_{l,n}$ being the computing time of the l th processor for *n*th iteration. We will illustrate a specific example about this in Section 4.

We point out that the relation scheme (10) in [8] is only the special case of the above algorithm when $\mu_{l,n} = 1$.

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

Algorithm 2.2 *Suppose that* $x_{0,r}$ *is the initial approximation of* $x(t_r)$ *at timepoint* r *and* $F_l(x_{n,r})$ *is satisfying (10). For* $n = 1,2\cdots,$ *until convergence, perform*

$$x_{n+1,r} = \omega \sum_{l=1}^{L} E_l F_l^{\mu_{l,n}}(x_{n,r}) + (1-\omega)x_{n,r}$$

with $\mu_{l,n} \geq 1, \quad \omega > 0, \quad rh \leq T$.

Next we will consider the more complicated situation, which is similar to Algorithm 2 in [6] and [10]. In this case the following new terminology is necessary. A sequence of sets P_n with $P_n \subseteq \{1,\cdots,L\}$ is admissible if every integer $1,\cdots,L$ appears infinitely often in the P_n , while such an admissible sequence is regulated if there exists a positive integer K such that each of the integers $1,\cdots,L$ appears at least once in any *K* consecutive sets of the sequence. Assume that $\{P_n\}$ is admissible and regulated, then we can get the following algorithm.

Algorithm 2.3 *Suppose that* $x_{0,r}$ *is the initial approximation of* $x(t_r)$ *at timepoint* r *and* $F_l(x_{n,r})$ *is satisfying (10). For* $n = 1,\cdots,$ *until convergence, perform*

$$x_{n+1,r} = (I - \omega \sum_{l\in P_n} E_l)x_{n,r} + \omega \sum_{l\in P_n} E_l F_l^{\mu_{l,n}}(Z_{n-k_l+1,r})$$

$$z_{n-k_l+1,r} = (x^1_{n-k(1,n),r}, x^2_{n-k(2,n),r}, \cdots, x^m_{n-k(m,n),r})^T$$

with $\mu_{l,n} \geq 1, \quad \omega > 0, \quad rh \leq T, \quad \phi \neq P_n \subseteq \{1,\cdots,L\}.$

3. CONVERGENCE OF THE ALGORITHMS

Before starting our convergence results concerning the above algorithms we should first introduce the following lemmas, which are necessary for the proofs of these results.

Lemma 3.1 *(See from [10]). If A is an H-matrix, then*
(a) $|A^{-1}| \leq \langle A \rangle^{-1}$;
(b) there exists a diagonal matrix P whose diagonal entries are positive such that AP is by rows strictly diagonally dominant, i.e.,

$$\langle A \rangle Pe > 0 \tag{11}$$

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

Lemma 3.2 *(See also from [10]). 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

$$\| P^{-1}M^{-1}NP \|_\infty < 1. \tag{12}$$

Lemma 3.3 *(See from [8]). Let matrices* $A = (a_{ij})$ *and* $B = (b_{ij})$ *be given with* $a_{ij} \leq 0$ *and* $b_{ij} \leq 0$ *for* $i \neq j$ *, and let A be an M-matrix. Then* $A \leq B$ *implies that B is an M-matrix.*

Lemma 3.4 *Let A be an H-matrix and let the matrix P be defined in Lemma 3.1. If the multisplitting* (M_l, N_l, E_l) *of A satisfies*

$$\langle A \rangle = \langle M_l \rangle - |N_l|, \quad l = 1,2,\cdots,L \tag{13}$$

then there exist a vector $u > 0 \in R^m$ *and a scalar* $\sigma \in (0, 1)$ *such that*

$$|hB_l N_l| u \leq \sigma u, \quad l = 1,2,\cdots,L,$$

where $B_l = (I + hM_l)^{-1}$.

Proof. By (13) we have

$$\langle hA \rangle = \langle hM_l \rangle - |hN_l|, \quad l = 1,2,\cdots,L, \quad h > 0.$$

Since *A* is an H-matrix, then <*A*> and <*hA*> are M-matrices. From Lemma 3.3, it derives that

$$\langle I + hA \rangle = \langle I + hM_l \rangle - |hN_l|, \quad l = 1,2,\cdots,L, \quad h > 0.$$

are also M-matrices. On the other hand $\langle M_l \rangle$ and $\langle hM_l \rangle$ are M-matrices. Using Lemma 3.3 we see that $\langle I + hM_l \rangle$ is an M-matrix. Hence, $\langle I + hM_l \rangle - |hM_l|$ is an M-splitting, for $l = 1,2,\cdots,L.$

Using Lemma 3.2 we obtain

$$\| P^{-1}\langle I + hM_l \rangle^{-1} |hN_l| P \|_\infty < 1, \quad l = 1,2,\cdots,L$$

and by Lemma 3.1 we have

$$\begin{aligned} &\| P^{-1}hB_l N_l P \|_\infty = \| P^{-1}(I + hM_l)^{-1} hN_l P \|_\infty \\ &\leq \| P^{-1} | (I + hM_l)^{-1} | hN_l | P \|_\infty \\ &\leq \| P^{-1}\langle I + hM_l \rangle^{-1} | hN_l | P \|_\infty \\ &< 1, \quad l = 1,2,\cdots,L. \end{aligned}$$

Let us denote $e = (1,1,\cdots,1)^T \in R^m$. For $l = 1,2,\cdots,L$, we then obtain

$$\begin{aligned} |hB_l N_l| Pe &= P(P^{-1} |hB_l N_l| P)e \\ &\leq \| P^{-1}hB_l N_l P \|_\infty \; Pe \\ &\leq \max_{1\leq l\leq L} \| P^{-1}hB_l N_l P \|_\infty \; Pe. \end{aligned}$$

We denote

$\sigma = \max_{1\le l\le L} \| P^{-1} h B_l N_l \mid P \|_\infty$ and $u = Pe$, (14)
then the required result is obtained.

Lemma3.5 *(See also from [8]). Consider a matrix* $B \in R^{m\times m}$ *with* $B \ge 0$ *and* $u \in R^m$ *with* $u > 0$ *be given, then* $Bu < u$ *implies* $\rho(B) < 1$.

Remark 3.1 *From the following proof for the convergence of the Algorithm 2.1, we will see that we only have to discuss the case of larger stepsizes, because we can achieve convergence without any condition on matrix A for sufficiently small stepsizes.*

Using the above Lemma 3.1-3.5, now we can prove one of our main results, which is a sufficient condition for the convergence of Algorithm 2.1.

Theorem 3.1 *Let* $A \in R^{m\times m}$ *be an H-matrix and let the matrix P be defined in Lemma 3.1. If the multisplitting* (M_l, N_l, E_l) *satisfies*

$$\langle A \rangle = \langle M_l \rangle - | N_l |, \quad l = 1,2,\cdots,L$$

then the sequence $\{x_{n,r}\}$ *generated by Algorithm 2.1 converges to the numerical solution of system (1) for initial vector* $x(0) = x_0$.

Proof. According to above Lemmas, we can complete the proof.

Theorem 3.2 *Suppose that the assumptions of Theorem 3.1 hold. Then the sequence* $\{x_{n,r}\}$ *generated by Algorithm 2.2 converges to numerical solution of system (1) for initial vector* $x(0) = x_0$ *when* $\omega \in (0, 2/(1+\sigma))$ *with* σ *satisfying (14).*

Proof. Similar to the proof of Theorem 3.1, we can also complete the proof.
Using the proof of Theorem 3.1 and [10, Theorem 2.8] we get our final result on the convergence of Algorithm 2.3.

Theorem 3.3 *Suppose that the assumptions of Theorem 3.1 hold. Then the sequence* $\{x_{n,r}\}$ *generated by Algorithm 2.3 converges to numerical solution of system (1) for initial vector* $x(0) = x_0$ *when* $\omega \in (0, 2/(1+\sigma))$ *with* σ *satisfying (14) and* $\phi \ne P_n \subseteq \{1,\cdots,L\}$.

4. CONVERGENCE SPEED COMPARISON OF SYNCHRONOUS AND ASYNCHRONOUS ALGORITHMS

We now consider the convergence speed comparison of the synchronous algorithm and corresponding asynchronous Algorism 2.1. For simplicity, we will only observe the $(n+1)$ st iteration of getting $x_{n+1,r}$ from $x_{n,r}$. We suppose that there exist one master and *L* processors in all. We see that in the synchronous case the master and all processors update only after the slowest processor (or that which has the most computational load) has communicated to the master. Also we assume that the following conditions hold:
1) The communication time is much faster than the computation time so that we can neglect the communication time.
2) In the cases of both synchronous and asynchronous iteration, all processors can keep the same computation speed.
3) The time required for updating of the master is constant and, without loss of generality, equals to one time unit, that is to say the processor which has the most computational load requires one time unit for computation.

We denote $\tau_{l,n}, l = 1,2,\cdots,L,$ the time required for l th processor at the $(n+1)$ st iteration. By letting $L = 5$, $\tau_{1,n}$=0.125, $\tau_{2,n}$=1, $\tau_{3,n}$=0.5, $\tau_{4,n}$=0.325, and $\tau_{5,n}$=0.25, in the case of asynchronous iteration, from $\mu_{l,n} = \lfloor \max_{1\le l\le L}(\tau_{l,n})/\tau_{l,n} \rfloor \quad l = 1,2,\cdots,L$, we obtain $\mu_{1,n} = 8, \ \mu_{2,n} = 1, \ \mu_{3,n} = 2, \ \mu_{4,n} = 3, \mu_{5,n} = 4$.
Clearly we see that there are many idle periods in the case of synchronous iteration, but in the case of asynchronous iteration there is none or the period is less.
On the other hand we now observe the convergence rates of fixed-point iteration . Let us denote

$$H = \sum_{l=1}^{L} E_l (h B_l N_l).$$

Since there exists a vector $u > 0 \in R^m$ and a scalar $\sigma \in (0,1)$ such that

$$| h B_l N_l | u \le \sigma u, \quad l = 1,2,\cdots,L,$$

then we have $\rho(h B_l N_l) < 1$.
It is clear that we obtain

$$\max_{1\le i\le m} \rho(H_i) = \max_{1\le i\le m} \rho(\sum_{l=1}^{L} E_l (h B_l N_l)^{\mu_{l,n,i}})$$
$$\le \rho\ (\sum_{l=1}^{L} E_l (h B_l N_l)) = \rho(H) < 1.$$

Hence we have got the conclusion that the convergence rate of the asynchronous iteration is geometric (for $\mu_{l,n,l} > 1$) and superior to that of the synchronous iteration.
As for Algorithms 2.2 and 2.3, since they are better than Algorithm 2.1 for some suitable relaxation parameters, we need not to compare the convergence speed of them with that of synchronous iteration.

5. REFERENCES

[1] A. Berman and R.J. Plemmons, Nonnegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.
[2] R. Bru, L. Elsner and M. Neumman, Models of parallel chaotic iterative methods, Linear Algebra Appl., 103(1988), 175--192.
[3] K. Burrage, Parallel and Sequential Methods for Ordinary Differential Equations, Oxford University Press, Oxford, 1995.
[4] D. Chazan and W. Miranker, Chaotic relaxation, Linear Algebra Appl., 2(1969), 199--222.
[5]]R. Jeltsch and B. Pohl, Waveform relaxation with overlapping splittings}, Rept. No.91--02, ETH, Zurich, Switzerland (1991).
[6] P. E. Kloeden and D. Yuan, Convergence of relaxed chaotic parallel iterative methods, Bulletin of Austral. Math. Soc., 50(1994), 167--176.
[7] 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.

[8] B. Pohl, On the convergence of the discretized multisplitting waveform relaxation algorithm, Applied Numerical Mathematics, 11(1993), 251--258.

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

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

[11] D. Bertsekas and J. Tsitsiklis, Parallel and Distributed Computation}, Prentice-Hall, Inc., Toronto, 1989.

Performance Evaluation of Distributed Computing

Guo Qingping[1] Guo Yucheng[2] Yakup Paker[3] Dennis Parkinson[3]
[1]Dept. of Computer Science and Engineering
Wuhan Transportation University, Wuhan P. R. China 430063
[2] Dept of Computer Science
Centre-South China Nationality University
[3] Dept of Computer Science
Queen Mary & West field College, University of London, E1 4NS
E-mail: qpguo@public.wh.hb.cn, paker@dcs.qmw.ac.uk

ABSTRACT

This paper proposes s network-computing performance based on that model. The paper points out that essential difference between network computing and MPP computing is their communication behaviours, the former is sequential and the latter concurrent. An uniform formula for system performance evaluation has been derived to cover those two major parallel processing systems. Furthermore a definition of parallel degree has been proposed and the distinction between system parallel degree and algorithm parallel degree has been analysed. Using the model this paper derives that for distributed network computing the speedup obeys the Amdahl's Law, and for MPP the Gustafson's modification has been verified. Furthermore some criterions of speedup, efficiency and granularity for network computing have been deduced and proved by measured results.

Keywords : Performance Analysis, Network Computing, Parallel Degrees

1. INTRODUCTION

At the middle of 80's, Professor C.A. Hoare suggested a Communicating Sequential Processing concept [1] to handle concurrency and parallel processing. Based on this concept the Britain has designed an OCCAM language, designed and manufactured Transputer chip for building memory distributed multi-transputer system [2]. The kernel idea of the CSP concept is using message-passing method for parallel processing.

In parallel processing researches, several paradigms have been tried in resent decades, including shared memory, parallel compilers and message passing. The message passing model has became the paradigm of choice because the many and variety of multiprocessing systems support it, as well as in terms of software system, languages and applications that use it.

Nowadays parallel processing system has two major developments: massively parallel processors (MPP) system and widespread use of network computing. However a common between distributed network computing and MPP is the concept of message passing.

In message passing paradigm there are two aspects and their relation playing important roles: computation and communication [3]. This paper analyses relations between computation and communication in PVM environment on network computing. Corresponding concepts such as speedup, efficiency, granularity and their behaviours in PVM network computing are also addressed. In section 2 features of PVM network computing are analysed and described by a formula. Section 3 compares analytical results with measured results for several double sizes increased calculated space nodes. Section 4 discusses speedup, efficiency and granularity. Difference of network computing and multiprocessor computing has been explored using the uniform performance formula. Section 5 summarises behaviours of PVM network computing. and points out pros and cons of PVM in network computing. A case study is given in section 6. Finally, section 7 gives conclusions.

Research supported by the UK Royal Society joint project (the Royal Society Q724) and the Natural Science Foundation of China (NSFC Grant No. 60173046).

2. FEATURES OF PVM ON NETWORK COMPUTING

PVM (parallel virtual machine) is an integrated set of software tools and libraries that emulates a general purpose, flexible, heterogeneous concurrent computing framework on interconnected computers of varied architecture [5]. Let us concentrate discussion on popular Ethernet LAN situation. In this environment communication between every pair of networked computers can be taken place if and only if there is no other use of the Ethernet. The communication in the LAN is not taking a point-to-point strategy, like the one in memory distributed multiprocessor system (e.g. multi-Transputer system), but a bus competition-based technology. In fact all the concurrent communications between corresponding computers have a sequential nature, which are taken place not simultaneously but sequentially.

Suppose the LAN is a homogeneous one for simplicity, an execution time of application can be represented like this:

$$T = \frac{A}{N} + C_2 N + C_3 + \varepsilon \frac{A}{N} \tag{1}$$

Where T is execution time, A represents a total calculation amount, N is number of PCs involved, so A/N represents a concurrent amount executed by each PC. C_2N describes the sequential nature of communication, C_3 is a constant initial latency, and $\varepsilon A/N$ represents a variable latency related to the concurrent amount. Obviously the formula (1) can be rewritten as

$$T = \frac{C_1}{N} + C_2 N + C_3 \tag{2}$$

Generally speaking, considering the execution time T as a function of N, the number of processors involved in a parallel processing, the T may be extended into following expression:

$$T = c_0 + \sum (c_k N^k + c_{-k} N^{-k})$$

here c_k is a independent coefficient of N. Obviously the equation (2) is main parts of the above extension, which

describes the nature of parallel processing.

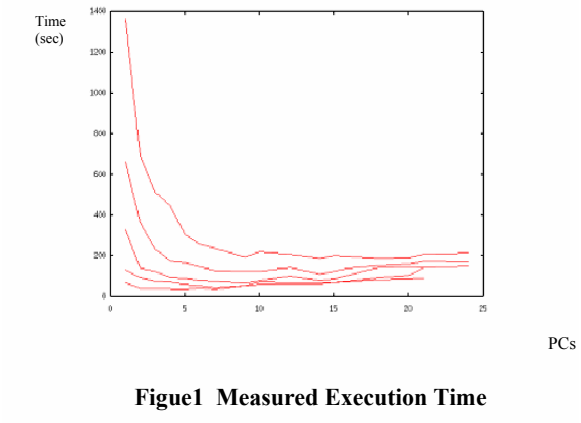
3. NETWORK COMPUTING PERFORMANCE MEASUREMENT

3.1. Network Structure and Measuring Method

The network we used is a local area network with30 PCs connected by an Ethernet in a big lab. Each PC has a Pentium 166 processor and 64Mbytes Edo RAM. The PVM version used is PVM3.3. In order to reduce interference from other users, we remote login to one of the 30 PCs in the middle of the night, then taking that PC as a host (master) which spawns number of slave processes running on corresponding slave PC in the same LAN. Obviously the best way of measurement would to isolate the LAN with outside network, severing any random traffic impacts from other networks, but this was not practical.

3.2 Algorithm Characters

As far as algorithm design is concerned there are several paradigms such as master-slave model (e.g. the processor farm), tree model, data decomposition and function decomposition. In the performance measurement we chose our recently developed a PVM version of Modified Tridiagonal Matrix algorithm, which uses an implicit method to solve cyclical temperature in ceramic/metal composites. This algorithm adopts a master-slave paradigm, using data decomposition methodology. The algorithm needs to divide transient time as several time steps and space distance in a cylinder as several space segments. Because the implicit method's nature we can chose big time interval and fix the number of time steps while changing the number of space segments, from the largest one to the smallest one. In fact the chosen space segments number series in the measurement is from 90720, the largest one to 5670, the smallest one, each time half smaller than the previous one. For each number of space segments running the program on the LAN, from one slave PC to the at most 24 PCs and measuring the execution time, we get a set of measured results, plotted in figure1. Using corresponding math approaches, e.g. the least square method, we determined the coefficients C_1, C_2 and C_3 in equation (2), and plot a set of theoretical execution time curves as shown in figure2. Figure3 plots the measured results and the theoretical results in one figure. It explicitly shows the formula (2) approaches the PVM environment behaviour in network computing.



Figue1 Measured Execution Time

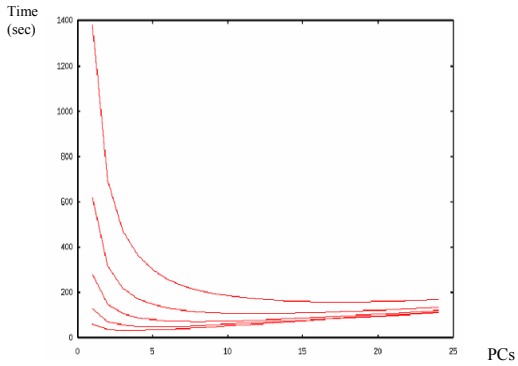


Figure2 Theoretical Execution Time

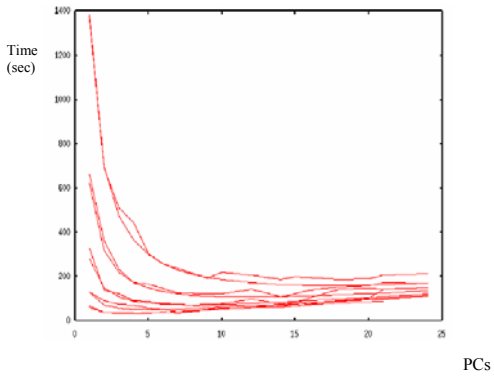


Figure3 Comparison of Measured and Theoretical Execution Time

4. SPEEDUP, EFFICIENCY AND GRANULARITY

4.1. Turning Point of Execution Time or Speedup Curves

Comparing program execution time on several slave PCs with one slave PC's we have got speedup. As can be seen from figure1 and 2 that the execution time of each case all has a turning point, assuming the N is continuous in equation (2) we can easy determine the turning point of execution time, as well as the speedup. In fact they are the same as:

$$N_{\max} = \sqrt{\frac{C_1}{C_2}} \tag{3}$$

Where C_1 mainly represents the computation amount and C_2 represents the sequential communication fact.

4.2 Difference between Network Computing and Multiprocessor Computing

Most advanced multiprocessor systems have employed a communication engine for message passing; therefore the sequential feature of communication in network computing now disappears and has been replaced by a separated concurrent communication mechanism. In this situation the formula (2) becomes

$$T = \frac{C_1}{N} + C_3 \tag{4}$$

Which has no extreme point. So the speedup can be written as

$$S = \frac{T_1}{T_N} = \frac{C_1 + C_3}{C_1/N + C_3}$$

Where C_1/N represents parallel portion of parallel algorithm, and C_3 represents the non-parallel portion of the algorithm. We assume the total execution time to be 1 for algebraic

simplicity, that is

$$\frac{C_1}{N} + C_3 = 1 \tag{5}$$

Then the speedup S can be represented as

$$S = C_1 + C_3 = N(1 - C_3) \tag{6}$$

This speedup is proportional to the number of processors. It is exactly the same result of the Gustafson's modification of Amdahl's Law for MPP machine's speedup [4]. From this point of view we can say the Gustafson's modification of Amdahl's law for speedup is only suitable for non-sequential communication situation. In that case, as problem scale increased the speedup is linear to the number of processors. In network computing environment, however, precondition of the Gustafson's modification has been vanished, and the speedup has a turning point, as shown in figure4, which can be determined by formula (3).

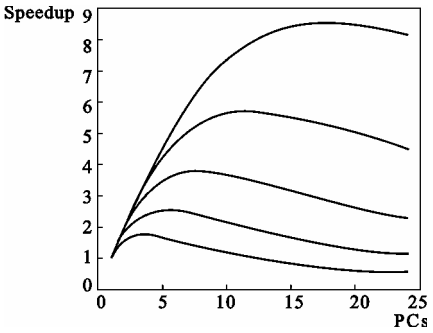


Figure 4 Speedup with Turning Point

4.3. Efficiency—a Magic Number of 0.50

A definition of processor efficiency is

$$E = S/N \tag{7}$$

So it can be yield that

Where S represents speedup. According to formula (2) the E can be written as

$$E = \frac{C_1 + C_2 + C_3}{C_1 + C_2 N^2 + C_3 N} \tag{8}$$

Replacing the N with a value of turning point, that is the equation (3), then the equation (8) becomes

$$E_{turnnng} = \frac{C_1 + C_2 + C_3}{2C_1 + C_3\sqrt{\frac{C_1}{C_2}}} \tag{9}$$

In general case we have that:

$C_1 >> C_2, C_3$

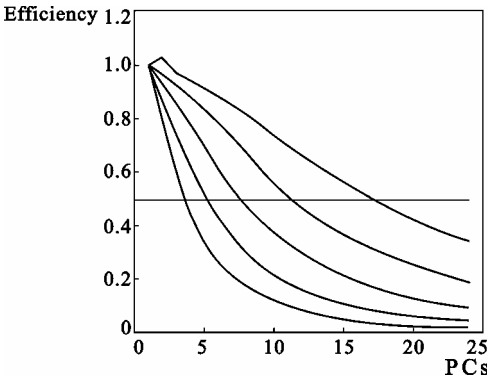


Figure 5 Efficiency

And because

$$\sqrt{C_1} >> \frac{C_3}{\sqrt{C_2}}$$

$$C_1 >> C_3\sqrt{\frac{C_1}{C_2}}$$

Therefore from equation (9) we can get

Figure (5) plots a set of efficiency curves. Comparing it with Figure (4) it is explicit shown that the speedup turning points are corresponding to the 0.50 efficiency.

There comes an important argument: if efficiency is reduced to 0.50, involving more computers in network computing makes no sense, because in this point more computers involved the performance will be reduced more, no further any gain.

4.4. Maximum Speedup

Maximum speedup is achieved at the turning point. Therefore

Consider $C_1 >> C_2$ and C_3, S_{max} can be written as

$$S_{\max} \approx \frac{1}{2}\sqrt{\frac{C_1}{C_2}} = \frac{1}{2}N_{\max} \tag{10}$$

That is a very useful relationship in PVM network distributed computing.

4.5 Granularity

It is clear that for PVM environments based on networks, large granularity generally leads to better performance. The lowest limit of granularity is

$$G_{\min} \approx \frac{C_1}{N_{\max}} = \sqrt{C_1 C_2} \tag{11}$$

Therefore quality poor network (C_2 is larger) needs bigger granularity; and big computation task (C_1 is larger) brings higher granularity.

$$S_{\max} = \frac{C_1 + C_2 + C_3}{C_1\sqrt{\frac{C_2}{C_1}} + C_2\sqrt{\frac{C_1}{C_2}} + C_3}$$

$$= \frac{C_1 + C_2 + C_3}{2\sqrt{C_1 C_2} + C_3}$$

As mentioned before the C_1 mainly represents a calculation amount of application and C_2 represents network communication nature, which is independent to applications. From formulas of N_{max} and G_{min}, that is equation (3) and (11), the following relations can be easy derived:

Therefore the network communication feature, i.e. the C_2, can be determined from application experience. If computation amount of other applications, i.e. the C_1, can be predicted, then the G_{min} and N_{max} as well as behaviours of those applications can be predetermined.

5. PARALLEL DEGREE

From equations (3) and (10) it is clear that the ratio of computation versus communication, that is C_1/C_2, plays an important role in distributed parallel computing. Therefore it is useful to give following definition:

Definition 1: General Parallel Degree (GPD)

The general parallel degree of an algorithm implementation is the ratio of total computation time versus the average

$$C_1 = N_{max} \cdot G_{min} \quad (12)$$

$$C_2 = \frac{G_{min}}{N_{max}} \quad (13)$$

communication time of a pair of computers. That is the GPD = C_1/C_2

In fact the C_1 can be rewritten as $C_1 = C_{11} \cdot C_{12}$ and the C_2 can be rewritten as $C_2 = C_{21} \cdot C_{22}$. Here the C_{11} represents the total number of computations and C_{12} denotes the elapsed time of each computation. Similarly the C_{21} represents average amount of communications of a pair of computers and C_{22} denotes the elapsed time of each communication. Therefore the GPD can be rewritten as

$$\begin{aligned} \text{GPD} &= C_{11} \cdot C_{12} / C_{21} \cdot C_{22} \\ &= C_{11} / C_{21} \cdot C_{12} / C_{22} \end{aligned} \quad (14)$$

It is obvious that the C_{11} / C_{21} is algorism dependent and the C_{12} / C_{22} is system dependent. Hence the following definition can be given.

Definition 2: System Parallel Degree (SPD)
The system parallel degree is the ratio of elapsed time of each computation versus elapsed time of each communication in each processor or computer, that is C_{12} / C_{22}. The SPD is system dependent and represents hardware characteristics as well as their performance.

Definition 3: Algorithm Parallel Degree (APD)
The algorism parallel degree is the ratio of total number of computations versus average amount of communications of a pair of computers, that is C_{11} / C_{21}, in a given distributed parallel algorism. The APD is algorism dependent and represents the algorism characteristics and its performance.
Therefore the GPD can be written as

$$\text{GPD} = \text{APD} \cdot \text{SPD} \quad (15)$$

From section 4 it is clear that the maximum speedup, the optimum number of nodes involved in parallel computation are all determined by the GPD. Therefore it is clear that

$$S_{max} = 0.5 \cdot N_{max} = \sqrt{GPD} = \sqrt{APD \cdot SPD} \quad (16)$$

From above discussion some conclusion could be made that an efficient distributed parallel algorism should be measured by APD, the algorism parallel degree; and performance of a distributed parallel system should be measured by SPD, the system parallel degree, this requires each CPU in a multiprocessor system is as fast as possible and communication time between a pair of CPU as short as possible. The former is algorism designer's job and the latter is system designer's job. A good implementation of a parallel solution consists of two parts: a good parallel algorithm design implemented on a good distributed system.

6. A VBF METHOD IN CLUSTER COMPUTING

As we have shown the communication overhead has a very strong influence in network-based parallel processing, i.e. so called cluster computing. In order to reduce communication overheads in parallel multigrid method, we proposed a virtual boundary forecast method (the VBF in sort) in domain decomposition parallel computing [6]. The kernel idea of the VBF method are:

(1) Forecast virtual boundary values using historic values at boundary points by some methods, e.g. linear forecast method, 2-order polynomial forecast method, maximum value of 2-order polynomial forecast method, etc. on each sub-domain; In this stage no communications between sub-domains;
(2) Performing multi-grid calculation on each sub-domain separately, independently in parallel without any communications between sub-domains; In this stage every sub-domain solves its own independent boundary problem;
(3) Smoothing results on the finest grid of whole domain in a few cycles to reduce errors derived by (2), in this stage there are communications between corresponding sub-domains, and updating the virtual boundary values;
(4) Using converges criterion to determine a sub-domain calculation should be terminated or not. If it can not be terminated the procedure returns to the (1) step; otherwise a termination token is sent to the neighbour computers, informing them the calculation on the sub-domain is finished, and there are no more communications between it and neighbours; meanwhile results on the sub-domain are sent to master computer for final results assembling.
(5) The whole calculation is terminated when all sub-domains are finished their own computing.

In fact the virtual boundary value update happens at step (3) as well as step (1). The latter step is more important than the first one in some sense.
Above algorithm dramatically reduces communication overheads in parallel multigrid method, which is a main drawback in efforts for parallel multigrid method. In fact the VBF method reduces communication cost by adding a little cost for virtual boundary calculation. Obviously communication in a local network is time consuming, but calculation on each processor is much more fast. The VBF method can achieve better optimal granularity, more big optimal number of computers involved, therefore speedup calculation. The measured results perfectly confirm our theoretical results.

7. CONCLUSIONS

From above analysis we have achieved following arguments:
(1) The nature of PVM environment based on network is parallel calculation plus sequential communication, which is different to modern multiprocessor system (e.g. MPP) with separated communication engine.
(2) Speedup of network computing obeys Adamhl's law, but the MPP's can be described with Gustafson's modification. The difference of them is derived from their different communication features.
(3) In network computing PVM environment, the maximum speedup S_{max} is equal to the half of a optimum number of computers, that is the N_{max}. The maximum number of computers involved (N_{max}), and minimum amount of granularity (G_{min}) for an application are simply determined by formulas (3) and (11).
(4) The efficiency of a distributed parallel algorithm is measured by its algorism parallel degree; the performance of a distributed parallel system should be judged by its system parallel degree. The maximum speedup S_{max} and the optimum number of computers involved, that is the N_{max}, in an implementation of a distributed parallel algorithm are straight related to the APD and SPD, which are described by formulas (15) and (16).
All those amusing results have been verified by measured results in numerical computing with or without adopting the VBF method, based on a PC computer local area network.

8. REFERENCES

[1] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice Hall International Series in Computer Science, 1985. ISBN 0-13-153271-5 (0-13-153289-8 PBK)

[2] David May and Mark Homewood, *Compiling Occam into Silicon,* Proceedings of the 20th Annual Conference on Microprogramming, IEEE, 1987.

[3] Guo Qingping and 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] John L. Gustafson, *Reevaluating Amdahl's Law,* Chapter for Book, Supercomputers and Artificial Intelligence, Edited by Kai Hwang, 1988.

[5] Al Geist et al. *PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing,* The MIT Press, 1994

[6] Guo Qingping et al., *Optimum Tactics of Parallel Multi-grid Algorithm with Virtual Boundary Forecast (VBF) Method Running on a Local Network with the PVM Platform,* Science Press, China; Allerton Press, INC. USA. Accepted, 2000.

Supplementing the Well-Known Factory Pattern for Distributed Object-Oriented Systems

Markus Aleksy, Axel Korthaus
Department of Management Information Systems, University of Mannheim
D-68131 Mannheim, Germany
E-mail: {aleksy|korthaus}@wifo3.uni-mannheim.de

ABSTRACT

In distributed object-oriented systems where resource users and resource providers/managers live on different nodes in the network it is common design practice to implement the well-known Factory Pattern for enabling the client part of an application to create new instances of classes or components on the server side at runtime in order to use these objects subsequently.
In this paper, we describe variants of and supplements for the Factory Pattern which address some missing, but very important aspects of the problem domain at hand, especially concerning the (automatic) destruction of remote objects. We discuss advantages and disadvantages of the different solutions and finally present a flexible compromise. Some thoughts about the implementation effort for this enhanced variant of the Factory Pattern and a short summary will conclude the paper.

Keywords: Factory Pattern, Life-Cycle-Management, Distributed Garbage Collection, CORBA

1. INTRODUCTION

At present, several different techniques are known for the development of distributed and/or parallel applications. In this paper, we will concentrate on the Common Object Request Broker Architecture (CORBA) standard [5] which has become very widespread in the area of distributed object-oriented systems. Not only does CORBA provide independence from computer architectures, operating systems, and programming languages, but it also frees the user from being bonded with vendors of specific ORB products. The last benefit was achieved by the introduction of unique object references, called Interoperable Object References (IORs), and a standardized transmission protocol in CORBA 2.0.
In this paper, we have a close look at possible ways to enhance and to implement the well-known Factory Pattern for CORBA-based, distributed object systems, and we discuss potential benefits and disadvantages of the different approaches.

2. OBJECT CREATION AND DESTRUCTION IN DISTRIBUTED SYSTEMS

During the development of an application that is based on distributed objects, it is usually important to provide the client with possibilities for creating and destroying server objects. Typically, a CORBA-based client would use the functionality of the CORBA Life Cycle Service
[6] for this purpose. This service, which was defined as part of the set of so-called CORBAServices, specifies interfaces for creating, destroying, copying, and moving CORBA objects. It contains the definition of the following three interfaces:

- Interface LifeCycleObject defines operations for copying, moving, and destroying CORBA objects.
- Interface GenericFactory defines a simple create_object operation. Every CORBA object can be created by this kind of Factory.
- Interface FactoryFinder finally defines operations that help find a specialized Factory for the creation of objects of a specific type.

The main disadvantage of the Life Cycle Service today is merely of practical nature, namely the lack of available implementations: hardly any CORBA ORB vendor offers products that implement this service. So, in order to work around this nuisance, most of the CORBA developers make use of the Factory Pattern which specifies a suitable alternative to the Life Cycle Service.
The Factory Pattern is used to solve the problem, that a client in a distributed object-oriented application does not know any implementation details of an object he wants to use, such as its implementation class, so that he cannot simply instantiate this class, e.g. by calling new Class_Name(); as it can be done in a non-distributed Java program.
However, a Factory should not be restricted to the task of creating new objects, but should also be responsible for monitoring and destroying objects. Otherwise, technical problems such as a breakdown of the client, the interruption of a network connection, an unsuitable implementation of the client, or an error caused by a wrong user behavior might lead to a situation where a resource that has been created and allocated on the server due to a message from the client will still continue to load the server, although it is not needed any longer and should be destroyed. At this point, it has to be mentioned that the current CORBA standard does not specify any mechanism for distributed garbage collection.

3. FACTORY APPROACHES

One possible solution to the problem mentioned above can be the coupling of the lifetime of a CORBA object with the existence of its creator, i.e., the client who triggered the remote object's creation (Observer-based Factory). The Notification-based Factory approach adds to this solution the following feature: in this approach, the creator is notified when the server object is to be destroyed, thus giving it the chance to veto the deletion plan. Another solution to the problem of efficient resource management is the so-called Evictor Pattern ([1], [3]). Its core technique is an efficient resource management based on monitoring. It assumes that resources least recently used (LRU), least frequently used (LFU), etc. are good candidates for automatic deletion. Yet another approach could be based on the concept of "leasing" resources. With the Jini specification [7], at the latest, this concept has become very popular. Jain and Kircher [4] even speak of a "Leasing Pattern" and describe the details of the problem domain, the structure and possible variants of that pattern. The basic principle is that a resource must not be left to its user infinitely,

but for a limited period of time only. When the specified period of time is over, several alternative actions can be performed.
In the following sections, advantages and disadvantages of these approaches will be discussed in detail. Techniques will be presented that can be used to enhance the classic Factory Pattern in order to increase the coupling of the CORBA objects' lifetimes with their creators and/or users.

State of the art
The state of the art variant of the Factory Pattern could be expressed with CORBA's Interface Definition Language (IDL) as follows:

```
module SOTAFactory
{
  interface Factory
  {
    Object create(in any criteria);
    void destroy(in Object ior);
};
};
```

The criteria parameter is of CORBA type any and might carry application-specific details such as the kind of object (*shared*, *unshared*) to be created.
A considerable disadvantage of this approach is that the created object cannot be deleted in case of a technical problem or implementation errors, thus needlessly continuing to occupy system resources.
The approaches mentioned before, i.e.,
- Observer-based Factory,
- Evictor-based Factory,
- Notification-based Factory, and
- Leasing-based Factory

can help to improve this situation. In the following subsections, they will be discussed in detail.

Observer-based Factory
This variant represents a first approach to the realization of an increased coupling of the creator of a remote object with the object itself. Here, the object creator has to provide the factory not only with general creation criteria, but also with its own object reference. At regular intervals, the latter is used by the factory object – or, alternatively, by the created object itself – in order to find out whether the creator is still alive or not. The following IDL interface might illustrate this approach:

```
module OBFactory
{
  interface Factory
  {
    Object create(in any criteria,
      in Object observed);
    void destroy(in Object ior);
}
};
```

A definition of an additional interface for the object under observation is not needed. To test whether the creator object is still existing, its operation _non_existent (inherited from CORBA Object) can be called.
The big advantage of this approach is the coupling of the created object's lifetime with the object creator's lifetime, so that no objects can survive that are not needed any more.
However, the following points might turn out to be detrimental:
- Probing the object creator's existence overly often can result in a relatively high network load.
- Prerequisite for this approach to work is that the object creator provides a CORBA object. Otherwise, there could be no callback and the _non_existent operation would not be provided.
- If it is necessary for the object creator to store the created object's IOR in a file in order to be able to terminate and reuse the IOR after a restart by restoring it from the file, this approach will probably fail, since the object might have been deleted already in the meantime.
- If the object creator passes the created object's reference on to another client and terminates, the created object will also terminate, so that the second client won't be able to use it and will hold a "dangling reference".

The last problem can be solved, for example, by introducing an additional operation change_observed(in Object ior) which has to be used to inform the factory (and the created object, respectively) about the new client to be observed, because the object's existence is now no longer dependent on the original object creator's existence, but on the new client's lifetime.
However, this enhancement will only work in certain circumstances. Problems occur, if the object creator needs to pass the remote object's reference to more than one other clients, because this approach is restricted to the observation of a single client. A recurring invocation of operation change_observed might of course lead to an erroneous behavior on the server side, because the last client to be registered must not necessarily be the client with the longest lifetime. Of course, a solution to this problem could be to manage a list of observable client objects, analogously to the Observer Pattern described in [1]. However, it must be recognized that observation of numerous client objects increases the network load notably.

Evictor-based Factory
The central idea of the Evictor Pattern approach ([1], [3]) is an efficient resource management based on monitoring the objects that have been created. If additional resources are needed, objects are automatically deleted on the basis of a specific strategy, such as "least recently used" (LRU), "least frequently used" (LFU), or a weighted combination of both. In a CORBA-based environment, this approach could be realized best by using the functionality of the Portable Object Adapter (POA) [5].
The IDL interface is no different from the usual factory interface, because the additional functionality is realized implicitly and does not affect the factory interface as seen by the client.
The transparency to the client is also the main advantage of this approach. There is no need at all for the client to concern itself for the deletion of the remote object. Furthermore, it is not necessary to have a CORBA object on the client side, because no callbacks are needed. For pure client applications, this might reduce the development and maintenance effort significantly.
However, the following aspects might be disadvantageous:
- The client does not know when the object it created will be removed, and, thus, cannot take any countermeasures.
- The monitoring functionality produces overhead. Consequently, this approach is only advantageous if realized on the basis of the POA functionality.

Notification-based Factory

This approach is similar to the first one in that the object creator's IOR has to be handed over to the factory. The difference is that the creator object is not under observation, but it will be notified by the factory or the created object itself if the created object is to be removed. Therefore, a suitable resource management strategy on the server side, such as the Evictor Pattern, is needed. Thus, the creator object gets the chance to free the occupied resources orderly, or, if they are still needed, to indicate its demand and veto the deletion. The following IDL interface shows how such a kind of factory could be defined in CORBA:

```
module NOFactory
{
 interface NotifiedObject
 {
  boolean notify(in any message);
 };
 interface Factory
 {
  Object create(
    in any criteria,
    in NotifiedObject notified);
  void destroy(in Object ior);
};
};
```

The boolean result of the notify operation implemented by the client indicates, whether the client agrees or disagrees on the deletion.

One beneficial aspect of this approach as opposed to the Observer-based Factory is that the network load is significantly smaller, because the factory is not forced to check the state of the object creator on a regular basis. Another positive aspect of the Notification-based Factory is that the object creator, after being informed about the deletion plans for the created object, has the option to request a prolongation of the created object's lifetime which will be granted or not depending on the overall system load.

Negative aspects of this approach are similar to those of the Observer-based Factory approach and include the following:

- Prerequisite for this approach to work is that the object creator provides a CORBA object implementing interface NotifiedObject. Otherwise, it could not be called back for notification.
- If the object creator passes the created object's reference on to another client and terminates, some time later the created object will also terminate, because the original creator cannot be notified anymore, so that the factory (or the object itself) presumes the object is not needed any longer and can be removed. Consequently, the second client will hold a "dangling reference".

Like in the Observer-based Factory approach, the last criticism could be overcome by extending the Notification-based Factory's interface with an operation change_notified(in NotifiedObject notified). But, since the same problems might occur as in the first approach, this extension only makes sense for special situations.

Since the typical observation intervals are much shorter than typical notification intervals, it can also be beneficial to combine the Observer-based and the Notification-based Factory approaches, because a termination of the client could be recognized much earlier for the price of an increased network traffic.

Leasing-based Factory

The last variant of the Factory Pattern to be presented in this paper is based on the concept of "leasing". A factory with leasing functionality can be implemented in two different ways. First, it might contain the complete leasing features itself. The follwing IDL interface illustrates this design:

```
module Leasing
{
 typedef unsigned long period;

 interface LeasedObjectFactory
 {
  exception TimePeriodOutOfBound
    { };
  Object create(
    in any criteria,
    in period seconds)
    raises(TimePeriodOutOfBound);
  void destroy(in Object ior);
  void lease(
    in Object obj,
    in period seconds)
    raises(TimePeriodOutOfBound);
 void cancel(in Object obj);
};
};
```

The second alternative is to delegate part of the functionality to the "leased object". See the following listing of the corresponding IDL interfaces:

```
module Leasing
{
 typedef unsigned long period;

 interface LeasedObject
 {
  exception TimePeriodOutOfBound
    { };
  void lease(
   in period seconds)
   raises(TimePeriodOutOfBound);
  void cancel();
  };

 interface LeasedObjectFactory
 {
  LeasedObject create(
    in any criteria);
  void destroy(
    in LeasedObject ior);
 };
  };
```

There are several advantages of the Leasing-based Factory solution. First, it is no longer problematic to hand over the created object's reference to other clients. Besides, the approach allows the implementation of light-weight client applications which do not need to provide CORBA objects. Furthermore, the client knows exactly, how long the created object will be available and when it will be destroyed.

The main disadvantages of this approach are the increased programming effort for implementing code that is responsible for the prolongation of leasing time, and the relatively high network communication overhead of this approach.

Table 1 Overview of the basic advantages and disadvantages of the different approaches

	Observation	Evictor	Notification	Leasing
Is it necessary to provide a CORBA object on the client side?	yes	no	yes	no
Can the created object's IOR be handed over to other clients without problems?	depends	yes	depends	yes
Does the object creator know (at least approximately) when the factory will destroy the created object?	yes	no	yes	yes
Can the object creator have an influence on the lifetime of the created object without affecting its own lifetime?	no	no	veto	yes
Does the approach cause an increased network communication load?	yes	no	relatively low	yes

Table 1 gives an overview of the most important pros and cons of the different solutions.

4. ALL-IN-ONE FACTORY

Depending on which problem is to be solved, one or the other of the presented solutions might be considered optimal. For example, if the client implementation has to be as simple as possible and the client has no special requirements on the lifetime of the objects it creates, the Observer-based and the Evictor-based approach would be sufficient, while the Leasing-based approach would not be advisable because of its complexity. On the other hand, the Leasing-based approach provides a very powerful resource management solution which leaves a great amount of determination to the client.
As a convenience, a factory that supports all of the approaches mentioned before at the same time could be implemented to let the developer handle only one factory component that gives him the freedom of choice, which kind of factory he needs for a specific application. The price for such an All-In-One Factory would be a much higher initial implementation effort, which, on the other hand, might turn out to be a big advantage afterwards. This is the case, if, for example, the factory component can be implemented to be very generic and to provide a plug & play interface.

5. CONCLUSION

In this paper, we presented several approaches to the implementation of an enhanced object factory in distributed object systems, and we discussed the pros and cons of the different approaches. If the developer knows the exact requirements of the application right from the beginning, he can decide for one specific factory variant. Otherwise, he should consider implementing an All-In-One-Factory, which is the most flexible solution. If it is designed and implemented to be generic, it might be reused in a great number of applications, thus paying off the initial investment. There might even be a potential for economies, because it is not necessary to maintain several different kinds of factories for different applications, if only the flexible All-in-One Factory solution is chosen.

6. REFERENCES

[1] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995): "Design Patterns – Elements of Reusable Object-Oriented Software"; Addison-Wesley Longman
[2] Henning, M., Vinoski, S. (1999): "Advanced CORBA Programming with C++"; Addison-Wesley Longman
[3] Jain, P. (2001): "Evictor"; Proceedings of 8th Patterns Languages of Programs Conference (PloP 2001), 11-15 September 2001, Allerton Park, Monticello, Illinois, USA
[4] Jain, P., Kircher, M. (2000): "Leasing"; Proceedings of 7th Patterns Languages of Programs Conference (PloP 2000), 13-16 August 2000, Allerton Park, Monticello, Illinois, USA
[5] OMG, CORBA/IIOP 2.5 Specification (2001), OMG Technical Document Number 01-09-01, http://www.omg.org/technology/documents/formal/corba_iiop.htm.
[6] OMG (2000): "Life Cycle Service Specification"; OMG Technical Document Number 00-06-18, ftp://ftp. omg.org/pub/docs/formal/00-06-18.pdf
[7] Sun Microsystems Inc. (2001): "Jini™ Architecture Specification"; December 2001, http://www.sun.com/jini/ specs/jini1.2html/jini-title.html

Robust Parallel Preconditioning Techniques forSolving General Sparse Linear Systems

Kai Wang and Jun Zhang
Laboratory for High Performance Scientific Computing and Computer Simulation,
Department of Computer Science, University of Kentucky,
773 Anderson Hall, Lexington, KY 40506-0046, USA
E-mail:kwang0@csr.uky.edu jzhang@cs.uky.edu

ABSTRACT

We develop a class of multistep successive preconditioning strategies to enhance efficiency and robustness of standard sparse approximate inverse preconditioning techniques. The key idea is to compute a series of simple sparse matrices to approximate the inverse of the original matrix. Studies are conducted to show the advantages of such an approach in terms of both improving preconditioning accuracy and reducing computational cost. Numerical experiments using one prototype implementation to solve a few general sparse matrices on a distributed memory parallel computer are reported.

1. INTRODUCTION

Consider a large sparse linear system of the form

$$Ax = b, \quad (1)$$

where A is a nonsingular general matrix of order n. A sparse approximate inverse preconditioning technique is first to find a sparse matrix M which is a good approximation to A^{-1}, then to solve a transformed system, in the form of

$$MAx = Mb, \quad (2)$$

by a Krylov subspace solver. The major driving force behind the search for efficient sparse approximate inverse preconditioners has been their potential advantages in parallel computing.

There exist several techniques to construct sparse approximate inverse preconditioners. They can be roughly categorized into three classes [2], sparse approximate inverses based on Frobenius norm minimization [6, 9], sparse approximate inverses computed from an ILU factorization [7, 12], and factored sparse approximate inverses [15]. Each of these classes contains a variety of different constructions and each of them has its own merits and drawbacks.

In this paper, we investigate a class of multistep successive sparse approximate inverse preconditioning techniques. A sequence of sparse matrices is computed inexpensively using an existing parallel sparse approximate inverse technique. The product of these sparse matrices is used to approximate the true inverse of the original matrix. Thus, instead of computing a costly high accuracy sparse approximate inverse preconditioner in one shot, we compute a series of inexpensive sparse approximate inverse preconditioners to achieve the effect of a high accuracy preconditioner. The sparsity pattern is adjusted when a new sparse approximate inverse matrix is computed.

2. ENHANCING ROBUSTNESS VIA SUCCESSIVE PRECONDITIONING

A sparse preconditioner may be computed using either dynamic or static sparsity pattern. The use of dynamic sparsity pattern may compute more accurate sparse approximate inverse preconditioners. But parallel implementation of dynamic sparsity pattern search can be quite expensive due to large amounts of data movement. It has also been noticed that high accuracy sparse approximate inverse preconditioners may be difficult and expensive to compute using a static sparsity pattern [5, 11]. Experimental results indicate that, compared to incomplete Cholesky factorization, sparse approximate inverse preconditioning wins only when the factorization is not required to be very accurate [11]. This is because it is difficult to determine a good static sparsity pattern a priori. Table 1 lists some test data using ParaSails, a software package implementing a sparse approximate inverse preconditioning, with different levels of sparsity patterns from Chow's paper [4]. It is to solve a symmetric positive definite matrix with n=12,205 and about1.4 million nonzeros.

Table 1 Parasails test results from[4]

Pattern	sparsity	iteration	setup time	solve time
A	0.25	754	2.0	39.3
A_2	0.47	539	40.0	33.7
A_3	0.80	243	491.0	20.4

We can see that use of higher level sparsity patterns, such as those of A^2 and A^3, does lead to better sparse approximate inverse preconditioners. This is indicated by the reduction in the number of preconditioned iterations (column 3). However, the CPU time in seconds needed to construct the preconditioners with higher accuracy (the setup time, column 4) increases substantially. The reduction in the solution time (column 5) does not compensate for the huge setup time. Hence, it is difficult to justify in this case to compute higher accuracy (more robust) sparse approximate inverse preconditioners.

We can approach the problem of choosing a suitable sparsity pattern in another way. Suppose a (*simple and inexpensive*) sparse approximate inverse preconditioner M_1 is computed for the matrix A, using any available sparse approximate inverse construction techniques, e.g., ParaSails with the sparsified pattern of A. If somehow we find that M_1 is not very efficient, we can compute another sparse approximate inverse preconditioner M_2 for the preconditioned linear system

$$M_1 Ax = M_1 b \quad (3)$$

We note that the systems (1) and (3) are equivalent, if M_1 is nonsingular as assumed. Thus, we compute another (*simple and inexpensive*) sparse approximate inverse preconditioner A_2 to the matrix $A_2=M_1A$. The combined preconditioner is then M_2M_1 for the matrixA. Here are a few comments to justify our successive sparse approximate inverse preconditioner in the form of product matrix M_2M_1.

- The computation of $A_2=M_1A$ can be done efficiently on parallel computers. If p is the average number of nonzeros in each row of A, the cost of computing A_2 is approximately equal to p folds of applying M_1 on a dense vector, or less than that of $p/2$ preconditioned iteration steps, assuming that M_1 uses the sparsity pattern of A.

Moreover, when we sparsify A_2 using a threshold parameter, the obtained sparsity pattern is more accurate than that of A^2, as it reflects the true pattern of A_2. The pattern of A^2 is computed from that of A using binary operations on the graph of A, without considering the size of the entries of A^2. Thus some useful information may get lost.

- If M_1 is an approximation to A, albeit not a very good one, then $A_2=M_1A$ tends to be closer to I than A does, or A_2 tends to be more diagonally dominant than A does. Thus, computing a sparse approximate inverse for A_2 is usually easier than computing one for A, given the same conditions.
- Intuitively the inverse A^{-1} of a sparse matrix A is dense. Then usually an accurate approximation for A^{-1} should be a dense matrix. This conflicts with our initial goal which is to find a sparse approximate inverse matrix M. However, if using the product of two sparse matrix M_2M_1 to approximate A^{-1}, we expect that M_2M_1 may be capable of holding more information than a single matrix M does. In this viewpoint, using M_2M_1 as a preconditioner is to some extent like using the factored sparse approximate inverse preconditioners [15].

A reader with recursive thinking will have already figured out the next step in the successive sparse approximate inverse procedure. If M_2M_1 is not good enough for preconditioning the matrix A in question, we compute a third sparse approximate inverse matrix M_3 for the product matrix $A_3=M_2M_1A$. This procedure can be continued for a few times to obtain a sequence of sparse matrices $M_1,M_2,\cdots M_l$, such that $M_l,M_{l-1},\cdots M_2M_1\approx A^{-1}$. If each M_i is good (but not necessarily very good) in some sense, we may expect that

$$\lim_{l\to\infty} M_l M_{l-1}\cdots M_2 M_1 = A^{-1}.$$

Because the matrix M_i becomes denser and denser as i increases, in each step we keep them sparse by dropping certain small size entries.

3. EXPERIMENTAL RESULTS

We conduct a few numerical experiments using a preliminary prototype code with multistep successive sparse approximate inverse techniques outlined in the previous section. This particular implementation uses ParaSails of Chow [5] as the backbone to build our multistep sparse approximate inverse preconditioner. For this reason, we refer to our preconditioner here as MultiStep with different steps. The code is mostly written in C programming language, with interprocessor communications being handled by MPI. The computations are carried out on a 32 processor (750MHz) subcomplex of an HP superdome supercluster at the University of Kentucky. Unless otherwise indicated explicitly, 4 processors are used in our numerical experiments.

In all tables containing numerical results, "n" denotes the dimension of the matrix; "nnz" represents the number of nonzeros in the sparse matrix; "np" is the number of processors used; "iter" shows how many iterations it takes for the preconditioned GMRES (50) to reduce residual norm by 8 orders of magnitude. We also set an upper bound of 5000 for the GMRES iteration; a symbol "-" in a table indicates lack of convergence. Similarly, "sparsity" stands for the sparsity ratio. In MultiStep, this is the sum of the number of nonzero entries of each M_i divided by the number of nonzero entries of original matrix A. "setup" is the total CPU time in seconds for constructing the preconditioner; "solve" is the total CPU time in seconds for solving the given sparse matrix using the preconditioner; "total" is the sum of "setup" and "solve". "e_1" and "e_2" are two dropping threshold parameters used to keep the memory cost low in each step. The content in the parentheses following "PS" indicates the *a priori* pattern used in ParaSails, e.g., *PS* (A^2) means that we use the sparsity pattern of A^2. Similarly, we use "MS" to denote MultiStep, the number in the followed parentheses is the step number, e.g., *MS (2)* means a 2 step MultiStep preconditioner.

3.1 Test Problems

Here we introduce the test problems which will be used in our experiments. The right hand sides of all linear systems are constructed by assuming that the solution is a vector of all ones. The initial guess is a zero vector. Convection-diffusion problem. The two dimensional convection-diffusion problem

$$-u_{xx}-u_{yy}-10\,(\sin x\cos \pi y\, u_x-\cos \pi x\sin y\, u_y)=0, \tag{4}$$

is defined on the unit square. Here the so-called Reynolds number is 10. Dirichlet boundary condition is assumed, but the artificial right hand side mentioned previously is used. The equation is discretized by using the standard 5-point central difference scheme. The resulting matrix is referred to as the 5-point matrix.

A three dimensional convection-diffusion problem (defined on a unit cube)

$$u_{xx}+u_{yy}+u_{zz}+1000\,(p(x,y,z)\,u_x + q(x,y,z)\,u_y+r(x,y,z)\,u_z)=0 \tag{5}$$

is used to generate some large sparse matrices to test the implementation scalability of MultiStep. Here the convection coefficients are chosen as

$$\begin{aligned} p(x,y,z) &= x(x-1)(1-3y)(1-2z),\\ q(x,y,z) &= y(y-1)(1-2z)(1-2x),\\ r(x,y,z) &= z(z-1)(1-2x)(1-2y). \end{aligned}$$

The Reynolds number for this problem is 1000. Eq.(5) is discretized by using the standard 7-point central difference scheme [14]. The resulting matrices are referred to as the 7-point matrix.

Test matrices. We also use MultiStep to solve a few sparse matrices. The FIDAP matrices [1] were extracted from the test problems provided in the FIDAP package [8]. They arise from coupled finite element discretization of Navier-Stokes equations modeling incompressible fluid flows. The RAEFSKY matrices are from modeling incompressible flow in pressure driven pipe, and are available from the University of Florida Sparse Matrix Collection. [2] The other matrices are from the well known Harwell-Boeing sparse matrix collection.

3.2 Comparison of preconditioning MA and A

We first compare the difference between preconditioning MA and A using ParaSails. The purpose of the comparison is to show that the matrix MA usually is more attractive than the matrix A to be used to construct a sparse approximate inverse preconditioner, which implies that Eq. (2) may be easier to solve, compared to solving Eq. (1).

The test results listed in Table 2 are from solving the two dimensional convection-diffusion problem (4). The first column is the number of rows (unknowns) of the matrices. The results in the second and third columns are to solve $Ax=b$ using a sparse approximate inverse preconditioner with the sparsity pattern of A^2. The results in the forth and fifth

1. All FIDAP matrices are available online from MatrixMarket of the National Institute of Standards and Technology (http://math.nist.gov/MatrixMarket).
2. http://www.csis.ufl.edu/1 davis/sparse.

columns are to solve MAx=Mb, which can be divided into two steps. First we use the sparsity pattern of A to obtain a sparse matrix $M \approx A^{-1}$, then we solve $MAx=Mb$ using sparse approximate inverse preconditioner with the sparsity pattern of MA . In the experiments, we set the parameters "e_1 " and "e_2 " in ParaSails to be 0, so that it does not drop anything during the preprocessing and postprocessing phases [5]. This implementation makes the sparsity pattern of MA the same as that of A^2.

We can see from Table 2 that the number of iterations needed to solve the matrix MA are usually20% less than that to solve the matrix A directly. That means that, with the same sparsity pattern or preconditioner sparsity, preconditioning matrix MA can get better convergence results than preconditioning matrix A directly. This property motivates us to develop multistep successive sparse approximate inverse techniques.

3.3 Properties of MultiStep

In this subsection, we present results from a few numerical experiments to demonstrate some favorable properties of multistep sparse approximate inverse preconditioners.

Table 2 Comparison of preconditioning A and MA for solving the 5-point matrices

	$Ax=b$		$MAx=Mb$	
n	sparsity	iter	sparsity	iter
100^2	2.58	195	2.58	139
200^2	2.59	354	2.59	249
250^2	2.59	443	2.59	354
300^2	2.59	535	2.59	400
350^2	2.59	576	2.59	427
400^2	2.60	681	2.60	536
450^2	2.60	821	2.60	625
500^2	2.60	864	2.60	688

Diagonal dominance property. Fig. 1 depicts the relationship between the number of steps in constructing the multistep sparse approximate inverse preconditioner and the ratio of strongly diagonally dominant rows of A_i in solving a few sparse matrices using MultiStep. The number after the matrix name in Fig. 1 denotes the number of iterations to solve the given matrix using the multistep sparse approximate inverse preconditioner with 8 steps.

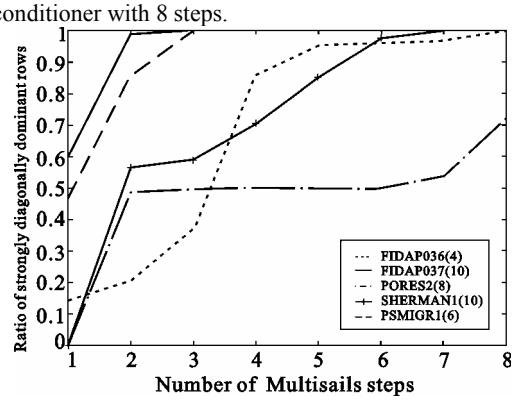


Figure 1 Relationship between the number of steps and the ratio of strongly diagonally dominant rows

From Fig. 1 we can see that when the number of steps increases, the ratio of strongly diagonally dominant rows of A_i increases quickly. In 4 of the test cases, the strongly diagonal dominance ratio finally reaches 1.0, i.e., 100% , after only a few steps. It is well known that diagonally dominant matrices are comparably easy to solve. So after 8 steps, all these matrices can be solved with the multistep sparse approximate inverse preconditioner in no more than 10 iterations.

In the experiments we also find that when the strongly diagonally dominant row ratio approaches 1, the structure of A_i tends to be similar to that of the identity matrix with many small offdiagonal entries, compared to the magnitudes of the main diagonal entries. It is possible to use a diagonal matrix to approximate the strongly diagonally dominant product matrix. So that we only need to compute the main diagonal entries of the last matrix and its inverse can be computed straightforwardly.

Sparsity ratio and iteration number. Table 3 gives some results from using MultiStep with different steps to solve the FIDAP031 matrix. We point out that if we use an oversparsified pattern of A to construct a preconditioner for A at the first step, the resulting preconditioner may not converge. However, this "poor" preconditioner can be used as M_2 in MultiStep as the basis to construct M_2, and M_2M_1 may make the preconditioned solver converge. In our situation, we think M_2M_1 may still not be good enough, because it converges in 1439 iterations. We then use M_2M_1 as the basis to construct M_3. In out tests, $M_3M_2M_1$ seems to be a good preconditioner for A, and it converges in 573 iterations. Continue doing this, we find that the multistep preconditioner converges in 342 iterations after 5 steps.

Table 3 Comparison of MultiStep ($\varepsilon_1=\varepsilon_2=0.03$) with different number of steps to solve the FIDAP031 matrix

steps	sparsity	iter	setup	solve	total
1	0.38	-	0.3	-	-
2	1.01	1439	1.7	2.5	4.2
3	1.43	573	4.4	1.8	6.1
4	1.62	392	8.0	1.4	9.4
5	1.72	398	14.2	1.7	15.9
6	1.78	342	19.5	1.8	21.3

Our other experiments also indicate that a larger step number leads to better convergence results. But it is not the case that the more steps in MultiStep the better the constructed preconditioner. This is because in each step we compute a matrix $M_i \approx A_i^{-1}$ and the memory cost of M_i will be counted into the whole memory cost of the preconditioner, as well as the construction cost. This obviously will increase both our computational cost and memory cost for MultiStep with a large number of steps. In Table 3 we notice that in the 6-step case, the preconditioner converges in 342 iterations, but the total computational cost is 5 times as much as that in the 2-step case. The reduction in the solution time does not compensate for the increase in the setup time. So unless it does not converge with a lower number of steps or in the case of solving one matrix with many right hand sides, usually we do not recommend too many steps in real applications, even though that may yield better convergence results. In Table 3, we think that 2 or 3 steps is a good compromise between reasonable computational cost and good convergence results.

3.4 Comparison of ParaSails and MultiStep

In Table 4, we give some comparison results between MultiStep and ParaSails for solving a few sparse matrices.

Table 4 Comparison of ParaSails and MultiStep for solving a few sparse matrices

Matices	Precondilioner	C_1	C_2	sparsity	iter	setup	solve	total
RAEFSKY 1 *n*=3242 *nnz*=294276	PS(A)	0.01	0.01	0.24	545	5.1	4.3	9.4
	PS(A^2)	0.02	0.02	0.38	148	210.5	4.1	214.6
	MS(2)	0.05	0.05	0.16	207	1.2	1.5	2.8
	MS(3)	0.02	0.02	0.44	50	10.2	0.2	10.5
RAEFSKY 2 *n*=3242 *nnz*=294276	PS(A)	0.01	0.01	0.68	786	6.0	7.5	13.5
	PS(A^2)	0.02	0.01	1.02	196	263.5	3.3	266.8
	MS(2)	0.05	0.02	0.32	535	2.9	3.3	6.2
	MS(3)	0.02	0.01	0.93	169	31.9	1.1	33.0
FIDAP024 *n*=2283 *nnz*=48733	PS(A^2)	0.0	0.0	4.86	-	9.8	-	-
	PS(A^3)	0.01	0.01	6.93	285	52.7	3.3	55.9
	MS(2)	0.001	0.002	4.47	799	12.2	4.4	16.6
	MS(3)	0.01	0.01	4.87	188	14.4	1.8	16.3
FIDAP028 *n*=2603 *nnz*=77653	PS(A^2)	0.0	0.0	4.29	789	19.3	6.7	25.9
	PS(A^3)	0.001	0.001	4.16	835	19.5	7.8	27.3
	MS(2)	0.004	0.004	2.97	255	13.4	2.9	16.2
	MS(2)	0.005	0.005	2.75	330	10.7	3.3	14.0
FIDAPM08 *n*=3876 *nnz*=103076	PS(A^2)	0.0	0.0	5.21	-	37.8	-	-
	PS(A^3)	0.0	0.0	12.91	-	377.4	-	-
	MS(3)	0.01	0.01	3.28	729	48.3	3.4	51.7
	MS(4)	0.01	0.01	5.12	291	142.2	2.2	144.5

We see that when constructing a preconditioner, MultiStep usually spends less time than ParaSails to reachthe same amount of sparsity ratio. According to our discussions in previous sections, the preconditioner computed from MultiStep is composed of a number of sparse matrices M_i. The memory cost of each sparse matrix is usually small and each of the sparse approximate inverse matrices can be computed very inexpensively. So the total computational cost of these sparse matrices is also small, compared with computing a single sparse matrix with comparable sparsity in the case of ParaSails.

Also the data in Table 4 show that with the same amount of memory cost (sparsity ratio), MultiStep usually has better convergence performance than ParaSails does. For solving the FIDAPM08 matrix, ParaSails does not converge when using either A^2 or A^3 as its sparsity pattern. However, MultiStep with a 3 step construction converges with a sparsity ratio 3.28.

3.5 Implementation scalability

The main computational costs in MultiStep are matrixmatrix product and matrix-vector product operations. As it is well known [10], these operations can be performed in parallel efficiently on most distributed memory parallel architectures.

Table 5 Scalability of MultiStep ($\varepsilon_1=\varepsilon_2=0.05$) for solving a 7-point matrix with $n=100^3$

np	sparsity	iter	setup	solve	Total
4	1.74	288	1953.4	232.8	2186.1
8	1.74	288	984.1	121.1	1105.0
16	1.74	288	501.9	44.4	546.3
24	1.74	288	361.7	29.4	391.1
32	1.74	288	281.8	24.7	306.5

The implementation scalability is tested using a three dimensional convection-diffusion problem (5) with the 7-point standard central difference scheme [14]. We let the matrix dimension to be 100^3. The nonzero number is 6940000. The matrix is solved by using a 2-step MultiStep. Table 5 shows the computational results with different numbers of processors. We can see that the Multi- Step preconditioner scales very well in this test case. In particular, we point out that the number of iterations remains to be the same in the test, when the number of processors incenses from 4 to 32. This is different from the simple domain decomposition preconditioners whose iteration properties are usually affected by the number of processors (domains) involved [3].

4. CONCLUDING REMARKS

We have proposed a class of multistep successive sparse approximate inverse preconditioning strategies for solving general sparse matrices. A prototype implementation named MultiStep is tested to show favorable convergence properties and computational efficiency of this class of new preconditioning strategies. The performance of the MultiStep preconditioner is indeed as good as what we expected. But some detailed work still needs be done in the future work, in order to build a software package that may be used in realistic large scale scientific computation and computer simulations.

Finally, we remark that the concepts of multistep successive preconditioning can be applied to other preconditioning techniques. It is also possible to construct multistep successive ILU preconditioners [13], or to construct multistep hybrid successive preconditioners using several different preconditioning techniques in different steps.

5. ACKNOWLEDGMENTS

The research work of the authors was supported in part by the U.S. National Science Foundation under grants CCR-9902022, CCR-9988165, CCR-0092532, and ACI-0202934, by the U.S. Department of Energy Office of Science under grant DE-FG02-02ER45961, by the Japanese Research Organization for Information Science & Technology, and by the University of Kentucky Research Committee.

6. REFERENCES

[1] S. T. Barnard, L. M. Bernardo, and H. D. Simon. An MPI implementation of the SPAI preconditioner on the T3E. Int. J. High Performance Comput. Appl., 13:107–128, 1999.

[2] M. Benzi and M. Tuma. A comparative study of sparse approximate inverse preconditioners. Appl. Numer. Math., 30(2-3):305–340, 1999.

[3] X.-C. Cai, W. D. Gropp, and D. E. Keyes. A

comparison of some domain decomposition and ILU preconditioned iterative methods for nonsymmetric elliptic problems. Numer. Linear Algebra Appl., 1(5):477–504, 1994.
[4] E. Chow. Parallel implementation and performance characteristics of least squares sparse approximate inverse preconditioners. Technical Report UCRL-JC-138883, Lawrence Livermore National Laboratory, Livermore, CA, 2000.
[5] E. Chow. ParaSails Users' Guide. Technical Report UCRL-JC-137863, Lawrence Livermore National Laboratory, Livermore, CA, 2000.
[6] E. Chow and Y. Saad. Approximate inverse preconditioners via sparse-sparse iterations. SIAM J. Sci. Comput., 19(3):995–1023, 1998.
[7] A. C. N. van Duin. Scalable parallel preconditioning with the sparse approximate inverse of triangular matrices. SIAM J. Matrix Anal. Appl., 20:987– 1006, 1999.
[8] M. Engelman. FIDAP: Examples Manual, Revision 6.0. Technical report, Fluid Dynamics International, Evanston, IL, 1991.
[9] M. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. SIAM J. Sci. Comput., 18:838–853, 1997.
[10] V. Kumar, A. Grama, A. Gupta, and G. Karypis. Introduction to Parallel Computing. Benjamin/Cummings Pub. Co., Redwood City, CA, 1994.
[11] P. Raghavan, K. Teranishi, and E. Ng. Towards scalable preconditioning using incomplete Cholesky factorization. In Proceedings of the 2001 Conference on Preconditioning Techniques for Large Scale Matrix Problems in Industrial Applications, pages 63–65, Tahoe City, CA, 2001.
[12] Y. Saad. Iterative Methods for Sparse Linear Systems. PWS Publishing, New York, NY, 1996. [13] L. Wang and J. Zhang. A new stabilization strategy for incomplete LU preconditioning of indefinite matrices. Appl. Math. Comput., to appear, 2002.
[13] J. Zhang. An explicit fourth-order compact finite difference scheme for three dimensional convection-diffusion equation. Commun. Numer. Methods Engrg., 14:209–218, 1998.
[14] J. Zhang. A sparse approximate inverse technique for parallel preconditioning of general sparse matrices. Appl. Math. Comput., 130:63–85, 2002.

A Secure Transaction Framework For M-Commerce*

Liu Quan, Guo Zhiqiang, Xu Chao
The School of Information Engineering, Wuhan University of Technology, Wuhan, Hubai China
E-mail：qliu@public.wh.wb.cn

ABSTRACT

Mobile e-business technologies aim to ensure that applications using secure transactions are developed with a consistent user experience across multiple phones, access technologies and usage scenarios. Mobile phones are rapidly evolving into much more than a wireless telephone, It is transforming into a Personal Trusted Device (PTD),with the ability to handle a wide variety of new services and applications. This paper illustrate how these services can be built around the PTD and realized in the Mobile Portal Supportive System (MPSS).

Keywords: mobile commerce, end-to-end security, personal trusted device, WAP, WPKI

1. INTRODUCTION

Recently Mobile commerce has become very important. .Many companies have been trying to find business opportunities in mobile domain. By developing middle-ware technology like e-payment system or security, and providing contents like ring tone, location based service and online games, they are struggling to survive in mobile business area. The mobile user's expectancy is getting instant and easy access to specific information. In order to meet their needs, a service provide should consider carefully how its application could benefit a person "on the move" :typically, offering brief, exact, and quick information. It is important for them to understand how to choice a mobile technology for using appropriate techniques to provide more efficient and secure service. In our research of the PTD secure m-commerce system, we propose the development of a secure m-commerce framework based on the use of WAP1.2.

2. PERSONAL TRUSTED DEVICE AND WAP SECURITY MECHANISM

Personal Trusted Device (PTD)

A PTD employs a mechanism for user verification in order to verify the person to whom the PTD belongs. Only after successful user verification may the PTD be used for mobile transactions. User verification is executed by the Security Element (e.g.,a smart card that performs cryptographic operations only after receiving a PIN entered by the user) in the device. In order to access multiple services, the PTD uses a certificate database for service specific matter. It contains actual certificates or pointers to certificate locations (certificate URLs).

Wireless Transport Layer Security (WTLS)

WTLS was created in order to facilitate secure wireless transaction without the need for extensive processing power and large sums of memory. Meanwhile WTLS promotes the fast processing of security algorithms. This is accomplished by reducing protocol overhead and enabling increased data compression in comparison to that of traditional SSL solution. The resulting effect is that WTLS can conduct appropriate security within a wireless network. These advances allow portable wireless devices to communicate securely over the internet, in order to achieve the requirement of Privacy, Authentication and Denial-of-service protection.

WTLS Certificates

In recent years one of the most effective and accepted ways of providing secure authentication has been through the usage of digital certificates. WTLS uses two types of certificates (WTLS server certificates and WTLS client certificates).
WTLS server certificates, defined as part of WAP 1.1,are used to authenticate a WTLS server to a WTLS client (handset) and to provide a basis for establishing a key to encrypt a client-server session.

WTLS client certificates, defined as part of WAP 1.2,are used to authenticate a WTLS client (handset) to a WTLS server. Both of them are formatted as either X.509 certificates or mini-certificates.

Two Phase Security

Two-phase secure WAP conversation occurs in two stages. First, the transmission between the web server and the WAP gateway occurs over SSL. Second, the onward transmission of this message over the air interface to and form the WAP browser device is over wireless networks using WTLS. Essentially, the WAP gateway serves as a bridge between the WTLS and SSL. Because of this, end-to-end security cannot be established. Only if users (include content providers) completely trust the WAP gateway service provider, can it be possible to say there is end-to-end security between end clients and end content providers. Nevertheless, there is possibility of attack form outside.

End to End Security Model

A WAP server is similar to a web server, except that it uses the WAP gateway protocol instead of the HTTP protocol that web servers use. This allows the WAP server to communicate directly with the WAP phone without a gateway. Thus, a WAP Server can be used to achieve end-to-end security.

3. USAGE SCENARIO: MOBILE PORTAL SUPPORTIVE SYSTEM

System Description

As a VIM card issue, network operators act as an important role in the enabling of mobile commerce value chain and other services requiring strong security. Network operators, acting as trusted parties for service providers, need to convince their clients that customer registration and key generation processes fulfill the defined requirements. It is shown in Figure 1 that Mobile Portal Supportive System (MPSS) is a such system architecture that supports wireless operator to sustain their future wireless data business. With one single platform, operators can turn their wireless network into a wireless portal and integrate application service and appreciation service from other application service providers or content providers in a

seamless and manageable way.
MPSS itself is not an application platform, and its function is to provide a supportive framework to support the whole wireless data business operations, say from customer provision to billing to customer care etc. Moreover, MPSS is a wireless portal engine where applications or services from other ASP and ICP which are integrated in a seamless and manageable way can easily be hooked up onto the wireless portal.
The objective of MPSS is to address all these business related issues and let operators focus back no their overall business and marketing strategy. And if this is combined with a good business model, a wireless operator will become a "Wireless Services Supermarket". As network quality and capacity will no longer be the main criteria for subscribers to choose network operators, the data services and secure transactions that an operator can provide will be their main considerations. There is no doubt that MPSS system is a distinct advantage for an operators competing in the future wireless data market .In the wireless world, security concerns are even greater than on the wired Internet. Wireless Internet application can succeed only if all the players are confident that transaction cannot be fraudulently generated or altered, that transaction are legally binding and that the confidentiality of private information is adequately protected. Such confidence depends upon the public key infrastructure (PKI) technology. In the following section we'll focus on the secure transaction and wireless trust services between subscribers and MPSS operator.

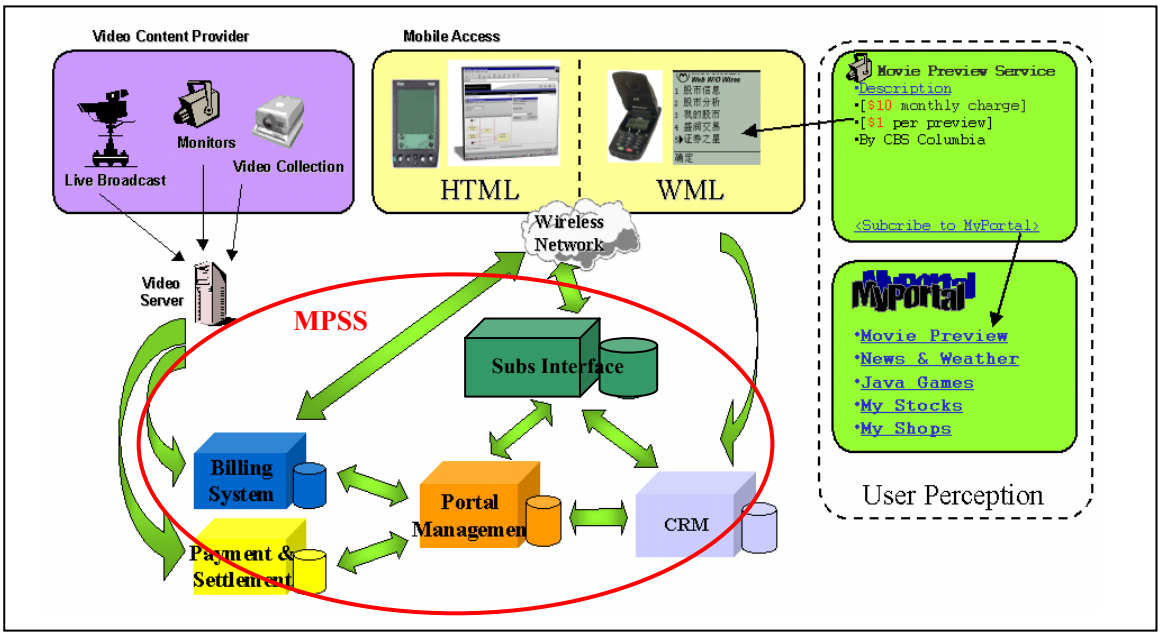


Figure 1 Mobile portal supportive system model

Implementation of Secure Transactions in MPSS
In MPSS, wireless clients need to use WTLS 3 species identification authenticates to strongly authenticate themselves to the WAP. This allows the WAP server to establish the identity of the user by means of the user service certificate related to the private key stored in the PTD. PTD authentication is established by the server's sending the PTD a challenge, which the PTD signs and returns. User verification is enacted by the entry of an access PIN known only to the user. This PIN allows access to the key pair used for user authentication. In the processing of two-phase secure WAP conversation establishment, the authentication is to the WAP gateway and not necessarily to the content server. WAP end-to-end transport layer security will allow authentication to a gateway located at the content server. In order for the user to access MPSS in securely ,a number of preliminary criteria must be fulfilled by both MPSS and user. The message flow for a successful login is shown in Figure 2.
MPSS must:
1) Offer WAP access to MPSS services through a secure WAP server.
2) Implement a sign-up mechanism by which the user registers for account access over WAP.
3) Issue a service certificate to the user for identification of the MPSS' s WAP service and authentication of the user to the MPSS. (This is a part of the registration process of the PTD.)
The user must:
1) Be in possession of a PTD that has the ability to perform authentication, establish secure sessions, create digital signatures and store certificates.
2) Initialize the PTD.
3) Register with the WAP MPSS service and receive a service certificate from MPSS operator.
The message flow for a successful transaction between user, content provider and MPSS operator is shown in Figure 3. According to the scenario in which a user accesses MPSS and browses to a WAP-enabled shop using his/her PTD, select goods and then pays for the goods with the PTD, core functions may be used for WAP shopping in MPSS environment. Figure 4 is a series of screens which are presented in order to show the purchaser of CD over the Internet the WAP device. In this commerce, depending on the payment agreement, verification of the signature may be performed either by MPSS or it may be passed to the payment system infrastructure. For pre-paid and post-paid users ,MPSS verifies the signature and charges user's MPSS account. In direct payment scenario, the user is prompted to select a bank account and enter the signature PIN. Upon receiving the PIN, the PTD signs the payment contract and return it, together with the user's bank certificate (not MPSS certificate). After receipt of the signed payment contract, MPSS equipment creates an authorization request message and sends it to the bank using a backbone network. The authorization request message includes, among other information, the signed

payment contract, providing proof of the user's intent to purchase.

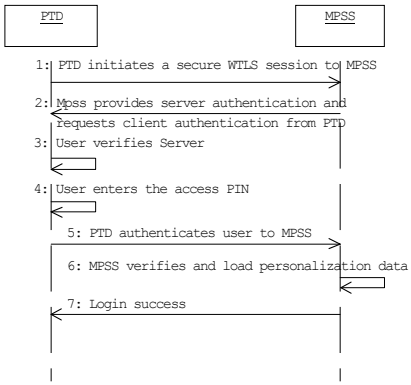


Figure 2 Successful login

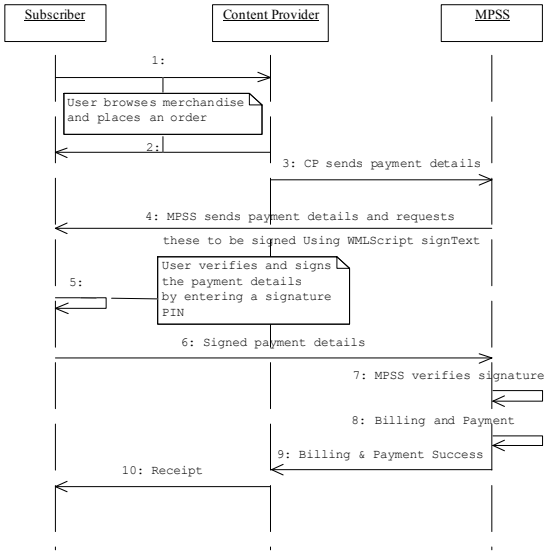


Figure 3 A successful transaction

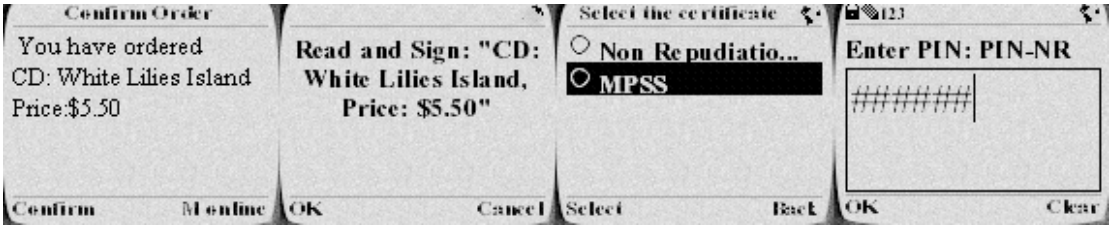


Figure 4 Interaction

4. CONCLUSION

The current username/password executions utilized in mobile commerce will greatly inhibit increased usage. WAP is an extraordinary revelation for the industry, and especially for the opportunities that exist for wireless communications. A WAP/WPKI-based system delivers authentication and signature capabilities that promise strong security. Based on such a prototype architecture, MPSS is still in the processing of being improve.

5. REFERENCE

[1] Mobile electronic Transactions. MeT PTD Definition. 21 February 2001.
[2] WAP Forum. Wireless Public Key Infrastructure Definition, 24-10-2000.
[3] NOKIA. Secure Identity in Mobile Financial Transactions. White Paper 2001.
[4] P. Dasgupta. N. Narasimhan. L. E. Moser. and P. M. Melliar-Smith. MAgNET: Mobile agents for networked electronic trading. IEEE Transactions on Knowledge and Data Engineering, 11(4) 509-525, 1999.
[5] W. Binder. Design and implementation of the JSEAL2 mobile agent kernel. In The 2001 Symposium on Applications and the Internet (SAINT-2001), San Diego, CA, USA, Jan. 2001.
[6] WAP Forum. WAP TM Transport Layer E2E Security Specification 17-08-2000.
[7] B. Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: John Wiley & Sons, 1994.
[8] Baltimore Technologies. Comparison of WPKI and SIM Toolkit, Mobile Internet Security. 2000
[9] VeriSign. Secure Wireless E-Commerce with PKI Verisign Inc. 2001.
[10] Kim Kwang Jo. Wireless Technology: Comparative Study on WAP. ICE525 Final Report. 2001.

An Algebraic Method for Verification of Arithmetic Program

Wang Jimin
Dept. of Computer Science, Peking University, Beijing 100871, China
Dept. of Computer Science, Lanzhou University, Lanzhou 730000, China
Lab. of Computer Science, Institute of software, CAS, Beijing 100080, China
E-mail: wjm@net.cs.pku.edu.cn
And
Li Lian
Dept. of Computer Science, Lanzhou University, Lanzhou 730000, China
E-mail: lil@lzu.edu.cn

ABSTRACT

This paper proposes an algebraic method to prove the correctness of Arithmetic Program which halts in finitely number of steps. The main routine is to simulate the program by a BSS computational model over the real numbers, thus it can be represented by a system of polynomial equations. The problem of proving program correctness will be translated into an algebraic one, which decides if the zeros of two systems of polynomial equations equals. The proof complexity of this method depends on the computational steps of a program.

Keywords: Program Verification, Arithmetic Program, BSS Model, Groebner Bases.

1. INTRODUCTION

It has been well recognized that program verification is important but difficult, because proofs of program correctness are complex and tedious. For programming logic in first-order or higher order logic and some kinds of programming language, some good and efficient methods have been made, especially, proving program correctness using formal methods have a great development in recent years[1, 2, 3, 4]. This paper explores a new method to verify program correctness for solving so-called arithmetic problem, which supplement existing program verification methods and might lead to new theoretical insights into this area. This method is based on the algebraic properties of BSS model of computation.

BSS model, defined by L. Blum et al. at the end of 1980s, is a model of machine working on an arbitrary commutative ring (or field) R (see [5, 6]). Their model has given rise to a whole new theory of computability and computational complexity. In the model, a machine has a finite control given by a finite graph, and an unlimited number registers, each capable of holding an element of R. The computational steps consist in computing polynomial and possibly deciding the next step by comparing the result of an evaluation with 0. A pair of integer registers can be used as points in order to retrieve and set any register. When R is a real closed field (real number field, e.g.), the computation of a BSS model which halts in finitely number of steps for its inputs can be represented by a system of polynomial equations, which called the algebraic property of BSS model. This model has a wider range of description than Turing machine, e.g., some problems like Mandelbrod set, Julia set and etc. can't be adequately addressed within the traditional discrete models of computation, but they can be described well by BSS model [6]. The research of BSS model and its applications has a fast and great development in the last few years [7, 8, 9].

We here restrict that the programs are so-called Arithmetic Program to solve arithmetic problems over a real closed field (see section 2). This is because any arithmetic problem can be represented by an algebraic system of equations in this kind of field.

The basic strategy of our method is as follows. First of all, a program P written by some high level programming language to solve an algorithm problem, is simulated by a BSS model. When P halts in the finitely number of steps, so does the model, thus a system of polynomial equations is obtained, which just describes the computational procedure. Finally, we check whether the same zeros of the two systems of polynomial equations or not (ignoring of all parameter variables); if yes, then the original program P is total correct to solve the arithmetic problem. In the way of Human-computer interaction, we do some examples of verification using the above method via the software system of Mathematica. The complexity of proofs using the method depends on the computational steps of a program.

The structure of this paper is as follows. Section 2 defines Arithmetic Problem and Arithmetic Program, and talks about the algebraic representation of Arithmetic Program. After an brief introduction to BSS model, Section 3 gives firstly the representation of a polynomial equations of the program, then discusses zeros property of the polynomial equations according to Groebner Bases of the ideal. In section 4 contains the steps of verification of Arithmetic Program using algebraic method, and implement the verification of a small practice example.

2. ARITHMETIC PROGRAM

Let $(R, +, *, 0, 1)$ be a (ordered) field, the *arithmetic terms* over R is defined as follows:
(1) Each constant over R and variable which its domain is over R is an arithmetic term.
(2) If p, q are constants, variables or rational polynomials over R, then $p+q$, $p-q$, $p*q$, p/q are arithmetic terms.
The *arithmetic formulas* over R is defined by the following statement:
(1) If p, q are two arithmetic terms over R, then $p=0$, $q=0$, $p=q$, $p \neq q$, ($p>q$, $p \geq q$, $p<q$, $p \leq q$ over an ordered field) are arithmetic formulas.
(2) If $p=0$, $q=0$ are two arithmetic formulas over R, then $p=0 \cap q=0$ and $p=0 \cup q=0$ are arithmetic formulas.
A problem is called Arithmetic Problem, if its solution can be described by the zeros of some arithmetic formula. For example, the problem of evaluating polynomial, $y = a_0+a_1x+a_2x^2+...+a_nx^n$ is an Arithmetic Problem.

A program is called an *Arithmetic Program*, if it is a program to solve arithmetic problem. For example, Straight-line programs, RAM programs and some programs written by some high level programming language to solve the problems of scientific computation.

Obviously, we can write an Arithmetic Program to solve the above problem of evaluating polynomial.

Property 1. Let K be a real closed field, and for any $x \in K$, the properties below hold.

$x \neq 0 \Leftrightarrow$ there exist $u \in K$, such that xu -1 =0

$x > 0 \Leftrightarrow$ there exist $u \in K$, such that xu^2 -1 =0

$x < 0 \Leftrightarrow$ there exist $u \in K$,such that xu^2 +1=0

$x \geq 0 \Leftrightarrow$ there exist $u \in K$, such that $x(xu^2 - 1) = 0$

$x \leq 0 \Leftrightarrow$ there exist $u \in K$, such that $x(xu^2 + 1) = 0$

$z = x/y \Leftrightarrow$ there exist $u \in K$, such that

$$\begin{cases} yz = x \\ yu - 1 = 0 \end{cases}$$

The proof of Property 1 can be found in many textbook of algebra, such as [10, 11].

Property 2. Any arithmetic problem over a real closed field can be represented by a system of polynomial equations over it.

Proof. For the arithmetic operation of two rational polynomials, after the least common denominator is found, they can be translated a rational polynomial. According to Property 1, its zeros will be equivalent to a system of polynomial equations.

For the relation operation of two rational polynomials, firstly, move over one side of the relation symbolic, and then use Property 1 to get a system of polynomial equations equivalently.

Let (I) $\begin{cases} f_1 = 0 \\ \cdots \\ f_r = 0 \end{cases}$ and (II) $\begin{cases} g_1 = 0 \\ \cdots \\ g_s = 0 \end{cases}$ be two systems of polynomial equations. The intersection of two systems of (I) and (II) is equivalent to a bigger system including all the equations of (I) and (II).

For (I)$\cup$(II), the equivalent system of equations is composed of these polynomial $f_i * g_j$ =0,where i =1, …,r, j =1, …, s, which is based on the property of polynomial ideal.

Example: Compute

$$y = \begin{cases} x^2 + 3 & \text{if } x \geq 0 \\ x - 1 & otherwise \end{cases}$$

This is an arithmetic problem over a ring of polynomial $R[x]$ in one variable, where R is real number field. And it is equivalent to the system of polynomial equations below.

$$\begin{cases} y - x^2 - 3 = 0 \\ x(xu^2 - 1) = 0 \end{cases} \text{ or } \begin{cases} y - x + 1 = 0 \\ xu^2 + 1 = 0 \end{cases}$$

Furthermore, it can be represented by a system of polynomial equations below.

$$\begin{cases} (y - x^2 - 3)(y - x + 1) = 0 \\ (y - x^2 - 3)(xu^2 + 1) = 0 \\ x(xu^2 - 1)(y - x + 1) = 0 \\ x(xu^2 - 1)(xu^2 + 1) = 0 \end{cases}$$

Remark 1:
Most of practical mathematical problems belong to the arithmetic problems over the real number field, and some arithmetic programs can be written by using various kinds of high level programming language to solve these problems.

3. SOME RESULTS ON BSS MODEL AND GROEBNER BASES

In [5, 6], two kinds of machine are defined: finite dimensional and infinite dimensional ones, the difference lying in the presence of an infinite number of registers in the input, output and state space, and of some machinery which is necessary in order to address such registers. We here will only introduce the finite dimensional model, because it can be proved that an infinite dimensional model is equivalent to an finite dimensional one in practical.

In the rest of the paper, the "field" will always mean "real closed field" although the original definition is a "ordered ring" R.

A *finite dimensional machine M* over R consists of three spaces: the input space $\overline{I} = R^I$, the output space $\overline{O} = R^m$ and the state space $\overline{S} = R^n$, together with a finite directed connected graph with node set $\overline{N}$ =1, 2, ..., N(N >1) divided in four subsets: *input, output, branch and computation nodes.*

Node 1 is the only input node, having no incoming edge and one outgoing edge; node N is the only output node, having no outgoing edge. They have associated with linear functions, mapping respectively the input space to the state space and the state space to the output space. Any other node $k \in$ 2, 3, ..., N -1 can be of the following types:

(1) A branch node k has two outgoing edges, giving out successors $\beta^+(n)$ and $\beta^-(n)$. There is a polynomial function $h_k : \overline{S} \to R$ associate to k, and for a given $\alpha \in \overline{S}$, branching on β^+(n) or $\beta^-(n)$ will depend upon whether or not $h_k(\alpha) \geq 0$.

(2) A computation node has a single outgoing edge, so that a next node $\beta(n)$ is defined, associated with it a polynomial map $g_n : \overline{S} \to \overline{S}$ .If R is a field then g_n could be taken rational.

We can view M as a discrete dynamical system over the full state space $\overline{N} \times \overline{S}$. M induces a computing endomorphism on the full state space:

$< 1, \alpha > \mapsto < \beta(1), \alpha >$

$<N, \alpha > \mapsto <N, \alpha >$

$<k, \alpha > \mapsto < \beta(k), g_k(\alpha) >$ if k is a computation node

$$<k, \alpha > \mapsto \begin{cases} < \beta^+(n), \alpha > & \text{if } h_k(\alpha) \geq 0 \\ < \beta^-(n), \alpha > & \text{if } h_k(\alpha) < 0 \end{cases}$$ if k is a branching nodes

The *computation of M* under input α is the orbit generated in the full state space by the computing endomorphism starting from < 1, I(α) >. If the orbit reaches a fixed point of the form

<N, β > for some $\beta \in \overline{S}$ we say that the machine halted, and that its output is $O(\beta)$.

Property 3. Any arithmetic program written by a high level programming language (Pascal Language e.g.) can be simulated in O(*n*) by a BSS model with finite dimensional, where n is the number of the instructions in *P*.

Proof. A BSS model is just like a flow diagram in programming language, so this is a simple procedure of simulation. For assignment instructions, this is only corresponding. The instructions of input and output will be translated respectively input and output nodes, a few of nodes must be added because BSS model requires that they have the property of unique one. As for branch instructions, they are become at least two nodes because the decidable condition is the first coordinate variable in the model. The complexity of simulation, therefore, is O(*n*).

Corollary 4. Let *P* be an arithmetic program, *L* a BSS model of computation. If P halts under some inputs in *T* steps, then *L* halts in O(*T*) for the same inputs.
If a finite-dimensional BSS machine over *R* halts in T steps, then its computation can be represented by the following polynomial equations [5].

$$(3.2)\quad \begin{cases} x_{k-1,1}(x_{k-1,1}u_{k-1}^2+1)(x_{k-1,1}u_{k-1}^2-1)=0 \\ \beta(n_{k-1}, x_{k-1,1}u_{k-1}^2)-n_k=0 \\ g_{(k)}(n_{k-1}, x_{k-1})-x_k=0 \end{cases}$$

where k =1, 2, ..., T, $u_{k-1,1}$ is a parameter variable, $\beta(n_{k-1}, x_{k-1,1}u_{k-1}^2)$ and $g_{(k)}(n_{k-1}, x_{k-1})$ are polynomials.
Groebner Basis is an important concept and technical tool in algebraic geometry. Let $I \subset k[x_1, \cdots, x_n]$ be an ideal, $I =< f_1, \cdots, f_s >$, the variable has the lexicographic order $x_1 > x_2 > \cdots x_n$. Then each $f_i \in k[x_1, \cdots, x_n]$ has a unique leading term LT(f), we denote by LT(I) the set of leading terms of elements of I. A Groebner Basis of I is a finite subset G ={ $g_1, \cdots, g_r$ }
if <LT(I) >=<LT(g_1), ⋯ ,LT(g_r) >.

Lemma 5. [12] Let G be a Groebner Basis of an ideal $I \subset k[x_1, \cdots, x_n]$, $f \in k[x_1, \cdots, x_n]$,Then f ∈I if and only if G reduce f to 0.

Elimination theory is a main method in solving a system of polynomial equations using Groebner Basis. The *l*th elimination ideal I_l of above ideal I is the ideal I ∩ k[x_1 , ⋯ , x_n]. Thus I_l consists of all consequences of $f_1 = \cdots = f_s = 0$ which eliminate the variables $x_1, \cdots, x_l$.

Lemma 6. (Elimination Theorem [12]) Let $I \subset k[x_1, \cdots, x_n]$ be an ideal and let G be a Groebner Basis of *I* with respect to lex order where $x_1 > x_2 > \cdots x_n$. Then, for every $0 \le l \le n$, the set $G_l = G \cap k[x_1, \cdots, x_n]$ is a Groebner Basis of the *l*th elimination ideal I_l .

A zero of elimination ideal I_l is said to be a partial zero of the original ideal I. The Extension Theorem gives a sufficient condition which can extend a partial zero to a complete zero of *I*. This theorem holds over an algebraically closed field. We talk below the extension of zeros over a real closed field *R*.
Let $I \subset R$[x, y, z_1 , ⋯ , z_n] is an ideal and let the variable order be x >y> $z_1 > \cdots > z_n$. $I =< f_1, \cdots, f_{n+1} >$, where

$$(3.1)\quad \begin{cases} f_1 = z_1 - \overline{f_1}(x) \\ f_2 = z_2 - \overline{f_2}(x, z_1) \\ \cdots \\ f_n = z_n - \overline{f_n}(x, z_1, \ldots z_{n-1}) \\ f_{n+1} = y - \overline{f_{n+1}}(x, z_1, \ldots z_n) \end{cases}$$

lemma 7. Let $I =< f_1, \ldots, f_{n+1} > \subset R[x, y, z_1, \cdots, z_n]$,where f_i have the above form. Then a zero (x_0, y_0) of the second elimination ideal I_2 can be extended to a complete one of I over real closed field.

Proof: These polynomial functions have special form, for each f_i ($0 \le i \le n$), both the degree and coefficient of leading term are 1 with respect to the variable order $z_1 > \cdots > z_n$,Thus, given a value of *x*, z_1 , ⋯ , z_n have a unique value respectively, and y is determined uniquely by x, z_1 , ⋯ , z_n . Furthermore, The elimination ideal I_2 is composed of those polynomials which only contains variable x, y. Based on the unique of y when x is given, a zero of I_2 will be extended to a zero of *I*. Therefore, the conclusion holds.
In previous (3.2), the u_{k-1} will be determined by x_{k-1}. We observe the second kind of polynomials,which β is a polynomial function about $x_{k-1}u_{k-1}^2$ .Combining with the first kinds of equations, we will find the β has nothing to do with u_{k-1} .According toLemma 7, the partial zeros of I, which only contains x_0 ,y and u_{k-1}, will be extended to a complete zero of *I*, where *I* is the ideal generated by all polynomials of the above equations (3.2).
For the previous example, the following is a PASCAL program, say P, to compute the original problem.

```
PROGRAM
VAR x, y: Real ;
BEGIN
READ(x);
IF x≥0 THEN y := x^2 +3
ELSE y := x -1;
WRITE ('y =',y)
END.
```

P halts in the 5th steps, and its corresponding BSS model can be represented by the following system of polynomial equations (simplification form).
$r_{1,0} = x$; $r_{1,1} = 0$ $r_{2,0} = r_{1,0}$; $r_{2,1} = r_{1,1}$; $r_{3,0} = r_{2,0}$; $r_{3,1}$

带格式的
带格式的
带格式的
带格式的
带格式的

=(x^2 +3)(n_2 -3) -(x -1)(n_2 -4); n_0 =1; n_1 =2; $r_{1,0}$ ($u^2 r_{1,0}$ +1)($u^2 r_{1,0}$ -1) = 0; y = $r_{3,1}$; $n_2 = 2(u^4 r_{1,0}^2 - u^4 r_{1,0}^2 + 2) + \frac{3}{2} u^2 r_{1,0} (u^2 r_{1,0} - 1)$

Let the order of variables be

$[r_{1,0}, r_{1,1}, r_{2,0}, r_{2,1}, r_{3,0}, r_{3,1}, n_0, n_1, n_2, y, x, u]$

Via computer algebra system Mathematica, we get a Groebner Bases of these polynomials in the above system of polynomial equations w.r.t. the order.

G = {-6

$-4u^2x - 2x^2 + u^2x^2 + 4u^4x^2 - u^2x^3 - u^4x^3 + u^4x^4 + 2y$,

$-8 - u^2x + u^4x^2 + 2n_2$, $-2 + n_1$, $-1 + n_0$, $-6 - 4u^2x$

$-2x^2 + u^2x^2 + 4u^4x^2 - u^4x^3 + u^4x^4 + 2r_{3,1} - x + u^4x^3$, $-x$

$+ r_{3,0}$, $r_{2,1} - x + r_{2,0}$, $r_{1,1} - x + r_{1,0}$ }.

4. MATHOD OF ARITHMETIC PROGRAM VERIFICATION

Here we use logic language to define the correctness of a program. Let Q be a problem, P a program to solve Q. P is said to be sound if, for given variables values, each output of P is a solution of Q. On the contrary, if for given variables values, each solution of Q is an output of P, then P is called completeness. So a program is correct if and only if P is sound and complete.

Suppose Q and P are characterized as two systems of polynomial equations, say (I), (II). All polynomials in (I)and(II) generate two ideals respectively, say <I>and <II>. From the view of algebra, <I > $\subset$ <II> means that P is sound, and <II>$_l$ $\subset$ <I >means that P is complete, where <II>$_l$ is an elimination ideal which only contains some variables occurring in <I>and a zero of <II>$_l$ can be extended a complete one of <II>. Therefore, P is correct if the two systems of polynomials have the same ideal which modular some parameter variables in <II>. On this grounds, we get the following result.

Theorem 8. Let Q be an given arithmetic problem, P be an arithmetic program to solve Q and P halt in finitely many of steps. The method below decide whether or not P solves Q correctly.

Step 1. Represent Q by a system of polynomial equations, say (I).

Step 2. Simulate P by a finite-dimensional BSS model that halts in the finitely many of steps.

Step 3. Represent the above derived BSS by a systemof polynomial equations, say (II).

Step 4. Compute the Groebner Bases respectively, say H and G, of two ideals generated by the polynomials in (I) and (II).

Step 5. Check whether or not all polynomials in H are reduced to 0 by G. If yes, then P is sound.

Step 6. Let M be the subset consisting all polynomials in G which only those variables in H occur. Check whether or not all polynomials of M are reduced to 0 by H. If yes, then P is complete.

Step 7. If P is both sound and complete, then claim that P solves Q correctly. Otherwise, P doesn't solve Q correctly.

Proof. "$\Rightarrow$" If P is correct, then solutions of arithmetic problem Q should be the solutions of program P, so H reduces all polynomials of M to 0, i.e., P is complete. Ignoring of the values of parameter variables, zero points of P should be those of Q, i.e. G reduce all polynomials of H to *0*, therefore P is sound-ness.

"$\Leftarrow$" The soundness of P guarantees that the solutions of P is those of Q when P neglecting parameter variables. Those polynomials of M are Groebner bases of the elimination ideal which eliminates all parameter variables in **G** (Elimination Theorem [12]), so the solutions of M is the partial solutions of G. Considering the special form of the algebraic representation of BSS model, which the lead term coefficient of parameter variables is 1, so the partial solutions can be extended to a complete solutions of P (Extension Theorem [12]). That is to say, M has a solutions, then P also has one and the solution of P is the extension solution. The completeness of P guarantees that H reduce all polynomials of M to 0, i.e., the solutions of Q are those of M. Furthermore, those solutions can also be extended to the solutions of P. So the program P is correct.

In the previous example, when the order of variables be [*y, x, u*], via the software system of Mathematica, we obtain the Groebner Bases H of all polynomials in the polynomial equations corresponding to the original arithmetic problem.

H = {$-x + u^4x^3$, -6

$-x - 4u^2x - x^2 + u^2x^2 + 4u^4x^2 - u^2x^3 + 2y$ }.

We let G reduce H, in step2, it is reduced to 0,so the program P is sound. In G, we investigate the set of polynomials M only concluding the variables y, x, u appeared in G.

$M = -6 - 4u^2x - 2x^2 + u^2x^2 + 4u^4x^2 - u^4x^3 + u^4x^4 + 2y$.

In step 1, that H reducing M is 0. Therefore, the program P is correct.

Remark 2 :

1.The procedure of practical verification, the computation of Groebner Bases has been done as a software package in most systems of computer algebra such as Mathematica, Maple and etc., so the testing speed is very fast under the real execution procedure.

2. In order to extend the range of verification of arithmetic program, we are exploring to use real RAM (Random Access Machine) program simulating the arithmetic program firstly. And then we simulate the RAM program by a BSS model. Where the real RAM means to extend RAM over integer numbers to real numbers, that is to say, the arbitrarily integer is instead of as an arbitrarily real number in the input-output tapes. This can dispose the lower machine instructions.

3. In the two procedures of an Arithmetic Problem and BSS model are presented by a system of polynomial equations, it is worthwhile to research how to correspond to the two kinds of parameters.

5. REFERENCES

[1] K.B. Helmut , W.E. Boebert, W.R. Franta, T.G. Moher. Formal Methods of Program Verification and Specification. Prentice-Hall, 1982.

[2] C. Zhang, R.A. Olsson, K.N. Levitt. Formal verification of a programming logic for a distributed programming language. Theoretical Computer Science, 1999, 216:213-253.

[3] D. Guaspari, C. Marceau, W.Polak. Formal

Ver-ification of Ada Programs. IEEE Trans. Software Eng. 1990,16(9):387-439.
[4] Z. Manna. Mathematical.Theory of Computation. McGraw-Hill, 1974.
[5] L. Blum, M. Shub, S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive function and universal machine. Bulletin of the American Mathematical Society, 1989,21:1-46.
[6] L. Blum, F. Cucker, M. Shub, S. Smale. Com-plexity and real computation. Springer, 1998.
[7] P. Koiran. A week version of the Blum, Shub & Smale model. In 36th IEEE symp. on Foundation of Computer Science, 1993,p486-495.
[8] S. Vigna. On the relations between distributive computability and the BSS model. Theoretical Computer Science, 1996,162:5-21.
[9] Wang Jimin, Li Lian. Two Applications of BSS Machine. The third Proceedings of Asian Symp. On Computer Mathematics, 1998.
[10] B.L. Vander Waerden. Algebra. Springer-Verlag, 1964.
[11] N. Jacobson. Lecture in abstract algebra (Vol 3): Theory of fields and Galois theory. Springer-Verlag,1964.
[12] D. Cox, J. Little, D. O'Shea. Ideals, Varieties, and Algorithms (second edition), Springer, 1996.
[13] A. Aho, J. Hopcroft, J. Uilman. The Design and Analysis of Computer Algorithms. Addison-Wesley Publishing Company, 1976.

Building Supercomputer with peer-to-peer Technologies

Alfred Loo,
Department of Information Systems, Lingnan University,
Tuen Mun, Hong Kong;
Email: alfred@ln.edu.hk
Y.K. Choi
Division of Computer Studies, City University of Hong Kong
Hong Kong
E-mail: dcykcho@cityu.edu.hk
Cyril Tse
Division of Computer Studies, City University of Hong Kong
Hong Kong
E-mail: dccyril@cityu.edu.hk

ABSTRACT

This paper presents inexpensive ways to implement a virtual supercomputer with peer-to-peer computers. A model with multiple servers will be discussed. It is possible to link millions of PCs in a peer-to-peer network. The capabilities of such network will be more powerful and reliable than any single supercomputer. Details for building such a peer-to-peer network are presented.

Keywords: Java, Intranet/Internet, Servlet, Web Server, peer-to-peer

1. INTRODUCTION

An anti-cancer research project begun in April last year allows people to join the fight against cancer with their home computers. This project is organized by Intel Corporation, the University of Oxford, the National Foundation for Cancer Research and United Devices. Its goal is to attract volunteers to contribute 24 million hours of computational time. Each volunteer simply downloads a program to his /her computer. The program runs when the computer is on but not being used, seeking to match ranges of potential cancer drugs to individual target proteins involved in cancer. Results from each computer are returned to the project coordinator via the Internet. To date, the project has been quite successful as over one million volunteers have installed the software. Despite the success, it will be extremely difficult for other organizations to migrate this peer-to-peer model to new and different projects.

2. DESIRED CHARACTERISTICS OF PEER-TO-PEER SYSTEMS

The cost of personal computers and workstations drops dramatically every year while the computing power increases continuously. These computers can also be used for general purpose applications (such as word processing, Internet browsing, etc.) when they are not being used for distributed computing. Using inexpensive personal computers and workstations to solve complex problems has increasing appeal [5].
In order to make the power of large p2p system assessable to small organizations or even individuals, the system must have the following characteristics:

1. We need the ability to initiate a program on a remote server from a client computer. Some software package must be installed on the servers and clients. It must be inexpensive (or even free) if we want to attract individuals to join the projects. There are a number of p2p products on the market and they are very expensive.
2. It must be easy to use.
3. It must be safe to use. The user does not need to examine every program from other users, which will be executed on his computer. Computers on the network might belong to different owners. There will be security risks (such as deleting files from your system) if you allow someone's programs to run on your computer.
4. It needs a minimum amount of user intervention. First time installation is acceptable, but the maintenance work should be reduced to minimum (or even no maintenance).
5. It should not rely on the product from a single vendor. If there is any existing product(s) which can achieve the task, we should not reinvent the wheel. We can simply extend the feature of these products. A model based on existing web server technologies can be used for p2p products and we will present this model in next section.
6. A client computer can reach a large number of power server computers within a short period.

The anti-cancer model is not able to meet point 3, 4 and 6. We will present a model which can solve the problems in the next section.

3. MULTIPLE SERVERS PEER-TO-PEER MODEL

Our model is presented in Figure 1. The client computer will invoke a Java application program which will divide a single task into many small sub-tasks. These sub-tasks are stored as queue on the hard disk of the system.
The application program will send a HTTP message to server which invokes a servlet [2] on the server. It will then transfer a sub-task to the server for further computation. (Note: There are many servers in the system but only one server is shown on Figure 1. We call them "power server" as they serve computing power to client.).
The server will then send the results to the client after the computation. The client will send another sub-task to the server if the sub-task queue is not empty. The client computer will collect answers of all sub-tasks.

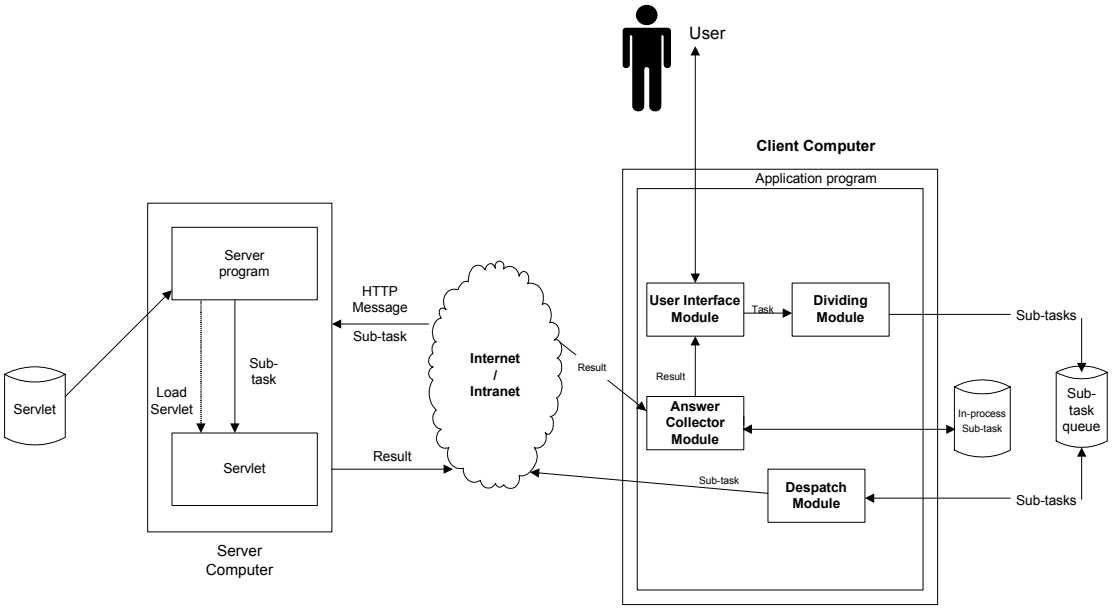


Figure 1 Multiple Servers peer-to-peer model

3.1 Advantages

Our model has the following advantages:

Security
Security is ensured by the "security manager" of Java. Owner can specify security policy [4] of each program. The security manager can monitor the activities of programs.

Simple Implementation
This approach is simple and easy to implement, which means that any experienced Java programmer should be able to work on peer-to-peer computing applications with limited training. Even individual with computer expertise can participate in this model.

Reasonably cheap set-up cost
The technologies described above are already available. There is no additional hardware and software cost incurred in implementing such a system.

Portability
The language Java used in this implementation is highly portable [1] so it can be run on any platform. Programs developed on one platform can be run without any modification on other platforms.

Robustness
The system can be programmed to detect a faulty computer and re-assign the sub-task to another computer to make the execution more reliable.

Flexibilities
All participants in the network can initiate a new project

Simple Maintenance
The maintenance job can be achieved by using the uploading function of most web servers. A special program on the client computer can automatically upload the new version of servlet to the web servers.

4. COMPONENTS OF THE MODEL

The computer programs will be divided into two major parts. One part is installed in the client computer while the other part is in the server computer.

In our model (Figure 1) we define the computer which serves CPU power as power server. We have only one client computer (i.e. the user's computer) and many servers (i.e. Computers which are willing to share their computing power). A performance evaluation of this model is available in [3].

4.1 Client Computer

There are four modules in the client computer.

4.1.1 Dividing Module

This module divides a complex job into many sub-jobs and store the sub-jobs in a queue.

4.1.2 Job Dispatch Module

This module (Figure 2) sends a HTTP message to invoke a java servlet program in the remote server through the Internet / Intranet. This module will establish a communication socket with the remote computer. It then picks up a sub-job from the queue and send it to the server.

After it dispatches a sub-task, it will also write a record to the "sub-task in-process" file. The record contains the sub-task, dispatch time and IP address of the remote powers. The system can detect a faulty power server computer by checking the dispatch time. In case the result of sub-task is not returned after a certain period, it will re-assign the sub-task to another power server.

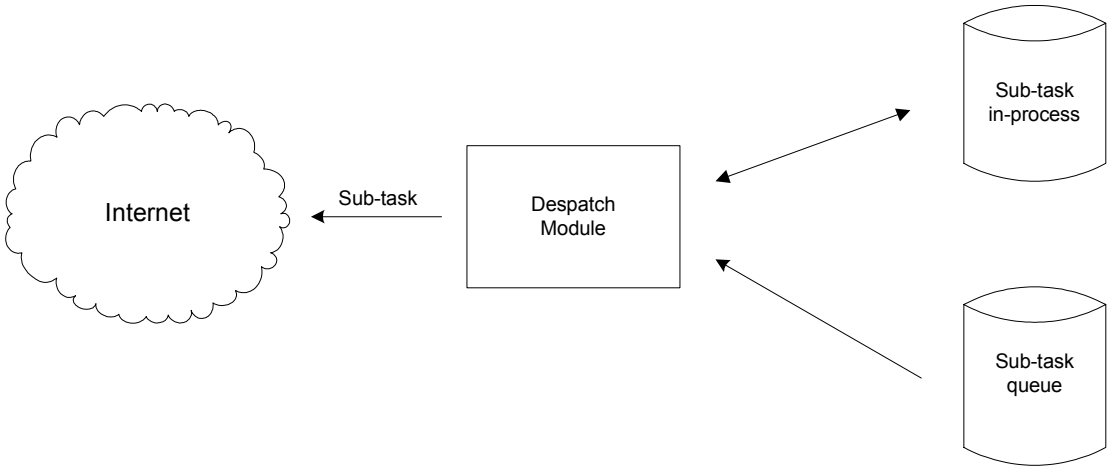


Figure 2 Dispatch Module

4.1.3 Answer Collector Module
This module collects answers from the remote computers and pass the answer to the local computer for further processing. The control will go back to the job dispatch module which will check if there is any sub-job in the queue. If the server cannot finish the job within a reasonable time (e.g. The server is down), this module will inform the "Job Dispatch" module to transmit the same sub-job to another server.

4.1.4 User Interface
This module will display information to user. It collects input from users and pass the processing request to the Dividing module.

4.2 Server Computers
The server computer receives a HTTP message from the client computers. It invokes the appropriate program which will handle the sub-task from the client computer. After it process the sub-task, it will send back the answer to the client computer.

4.2.1 Web Server
The web server will initiate the requested program after it receives a HTTP message from the client computer. It will also check the user identity if necessary.

4.2.2 Servlet
The servlet will first establish a communication channel with the client computer. It will then receive a sub-task and process it according. It will send back the answer to the "Answer Collector" module of the client computer. The servlet will check if the client still has any sub-task left. It will process the sub-task until the "Dividing" module of the client computer sends it an ending signal.

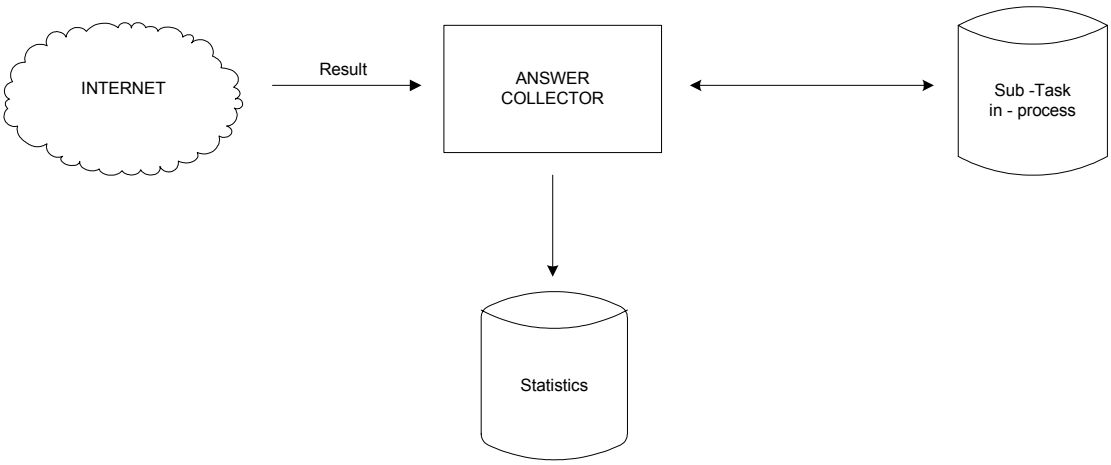


Figure 3 Answer Collector Module

5. ETHICAL CONSIDERATIONS

The successfulness of implementing the Java-based supercomputer with peer-to-peer network is highly depends on the cooperation and support of Internet users. Users should able to use their own free will to decide whether they wish to have their idle CPU power being shared to the others. However, due to the widely spreading of network viruses and Trojan programs, people become worry about download anything from the Net.

Apart from designing the servlet programs in a 100% safe mode, it is also necessary for the users to monitor the progress and control the running of servlets. When the servlet has already finished its calculation, it should terminate itself after returning the result.

Users should be able to stop the execution of the servlet program at any time. Another way to encourage people to donate their idle CPU power is to apply such supercomputer technology where the users can get benefits. News sites, entertainment sites, or software upgrade are good examples. All of these sites could trade some of the valuable contents for processing power on users' machines.

6. CONCLUSION

Our contribution in this paper is to bring several peer-to-peer technologies together to build a virtual supercomputer. The major advantage of our model is that all participants can initiate their own new project. Every participant will thus benefit from the system..

7. REFERENCES

[1] Campione, M. and Walrath, K., 1997, The Java Tutorial – Object Oriented Programming for the Internet, Addision Wesley.
[2] Hunter, J., 1998, Java Servlet Progamming, Oreilly.
[3] Loo, A., Bloor, C. and Choi, C., 2000, Parallel Computing using Web Server and Servlets, Internet Research, Volume 10, Number 2.
[4] Oaks, S., 2001, Java Security, O'Reilly.
[5] Oram, A., 2001, Peer-to-peer: Harnessing the Power of Disruptive Technologies, O'Reilly.

An Integrated Architecture for a Distributed Adaptive Learning Environment

Qiangguo PU*
Computer Center, University of Science and Technology of Suzhou
Suzhou, Jiangsu 215011, China
E-mail: hpu@bigfoot.com

ABSTRACT

The Centre for Computing and Information Systems (CCIS) at Athabasca University, Canada has been developing DALE- a Distributed Adaptive Learning Environment. This paper presents the concepts of distributed adaptive learning environments and the integrated architecture for DALE. The author was a senior visiting scholar at CCIS at Athabasca University. He took an active part in the project.

Keywords: agent technology, integrated architecture, distributed adaptive learning environment, web-based education, e-education

1. INTRODUCTION

The World Wide Web is a popular and useful instructional medium for a number of reasons. It is easily accessible, it supports flexible storage and display options, it provides a simple yet powerful publishing format and a means to incorporate multiple media elements. Interestingly, instructional effectiveness is not a proven characteristic for World Wide Web courseware and in many instances delivery via the WWW can impede rather than enhance learning when compared to conventional publishing forms.. Today, most people think of the World Wide Web (WWW) as an ideal environment for information publishers. Universities use it to disseminate administrative and marketing information to faculty, students, alumni and potential students. Commercial use of the WWW is growing phenomenally as companies, large and small, jump on the bandwagon to market their products and services. But the WWW, in combination with other Internet tools such as Usenet Newsgroups, Email and Telnet, can be an interactive learning environment as well. And the creative implementation of these tools makes the WWW an ideal environment for distributed learning.

The goal of Athabasca University (AU) (http://www.athabascau.ca) is to provide lifelong open learning (Race, 1994)[1] by reducing barriers to education. Generally students can take the courses at the time and place of their convenience. AU serves about 24,000 students a year. Most of the students are part-time. AU offers 420 undergraduate courses and 15 undergraduate degrees. The University also has four graduate degrees and serves about 1,000 graduate students annually. The Masters courses are paced with fixed start-dates, but most undergraduate courses can be entered at any time in the year, and students have six months to complete all the assignments and the final examination.

The Centre for Computing and Information Systems (CCIS) (http://ccism.pc.athabascau.ca) at AU offers four undergraduate credentials shaped primarily by ACM curricula, professional requirements, and student needs. All the credentials are somewhat more applied than a traditional Computer Science degree. Most of the 2000 undergraduate students are part-time, generating about 3,000 course registrations per year. The most recent credential is a Masters in Information Systems aimed at working IT professionals. It is anticipated that in two years the Masters in IS will have about 150 students per year generating about 540 course registrations per year.

AU students are distributed not only across Canada where AU is located, but also across North America and the rest of the world. In the past, the separation by space and time of the AU students diminished the effectiveness of distance education approaches. Informal peer support and group work efforts were particularly restricted, with students often feeling lost in their attempts to deal with new endeavors in isolation. Now computer technology provides a new and innovative approach to open learning. E-mail, computer based conferencing, structured hypertext and the virtual reality technologies change the nature and enhance the quality of distance education. CCIS has been leading AU in developing an online learning system. Starting in 1995 (Holt, Gismondi, Fontaine, and Ramsden, 1995) [2], CCIS converted 20 Computing Science courses from the traditional telephone/correspondence delivery to World Wide Web (WWW) delivery. Students at home and work are supported with e-mail, chat, and computer conferencing. Since then AU has become increasingly reliant on electronic delivery.

The Centre for Computing and Information Systems at AU has been developing DALE - a Distributed Adaptive Learning Environment. This paper presents the concepts of learning environments, adaptive learning environments and distributed learning environments, and introduces the distributed adaptive learning environment-DALE. The author was a senior visiting scholar at CCIS at Athabasca University. He took an active part in the project.

2. DISTRIBUTED ADAPTIVE LEARNING ENVIRONMENTS

We define the learning environment as consisting of all the interaction with materials, the tools, the interactions with peers, and the interaction with tutors, access to other materials, and even the approaches to learning available. In a distributed learning environment the various resources are geographically distributed but connected by information technology. The World Wide Web offers the most successful model of a distributed learning environment. Originally web based learning was restricted to basic to text and images presented through HTML. With the rapid increment of bandwidth available on the web these have been supplemented with audio, video, animations, and simulation. The web has become a virtual environment – a place where people work, play, shop, and learn.

* Qiangguo Pu is an associate professor of Computer Center at University of Science and Technology of Suzhou, China. He was a visiting scholar at McMaster University, Canada and a senior visiting scholar at Athabasca University, Canada. He is author or co-author of more than 40 publications (journal papers and conference proceedings). His research interests include the application of computers and other technology in distance education and control systems.

A basic premise of the CCIS distributed learning environment is that our learners are best served by an underlying distributed architecture with much of the processing occurring on the learner's computer. Advanced Internet gaming activities are based on such a distributed model. There are services that must be offered centrally but many functions are best performed locally (URL re desktop). A local component best provides an environment supporting learner autonomy, empowerment, and privacy. Learners can provide services to other learners in a peer-to-peer network ala Napster (www.napster.com). A distributed learning environment facilitates a learner-centered educational paradigm and promotes active learning. Our model of web-based distributed learning focuses on a model of guided self-discovery in which learners engage in learning activities at the time and place of their convenience. However, where it best fits the learners' needs we also use the technology a more traditional "paced" approach where the learners are part of a cohort group with a fixed schedule for all assignments etcetera. We also used hybrid models that fall between these two models.

The term adaptive learning environment refers to technology that will adapt the environment in various ways. First, CCIS wants its content to be reusable across various modules and courses. XML combined with international standards such as IMS (see www.imsproject.org) provides a standard way of storing learning objects making them available for use in a variety of ways. Second, content must be able to be delivered across different platforms. XML, XSLT, and related Java technologies provide the means for transforming material for presentation on various devices such as desktop computers and wireless handheld devices. Third, CCIS wants the content to be dynamically adaptable to the needs of particular learners. Intelligent software agents (Lin and Holt, 2001)[3] provide the key intelligence for a wide range of adaptations. Finally, standards such as IMS and a variety of tools help make material adaptable to the needs of the authors, instructors, and the educational institutions.

3. ARCHITECTURE FOR A DISTRIBUTED ADAPTIVE LEARNING ENVIRONMENT

CCIS academics have focused much of their research our Distributed Adaptive Learning Environment (DALE) project. The DALE project explores the issues in creating a learner-centered platform for electronic distance learning, particular for a computer and information systems programme. The DALE project is based on a distributed processing architecture to take full advantage of the powerful student client computer (Gelernter, 2000)[4] and investigate the potential of peer to peer computing (Oram, 2001)[5].

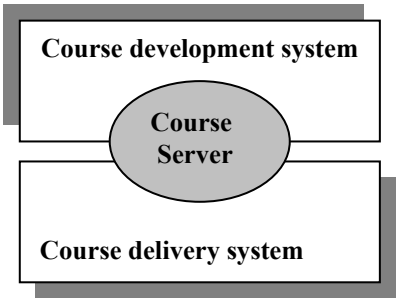


Figure 1 The whole picture - course authoring, course delivery and course server

It is important that there be an overall integrating architecture for organizing the various functions and the associated technology (see Figure 1). It must be capable providing a stimulating, interactive informative environment for situated learning, peer-to-peer collaboration, and communicating with tutors (DALE Research Team, 2002)[6]. It should support tutors facilitating student learning and support professors in designing, creating, and delivering their course materials. The development and delivery systems are both critical and enhances the functionality of both systems. For example, feedback from the delivery informs timely revision of course materials. However, this paper focuses on the delivery system and the student-learning environment.

Course Server

As can be seen from Figure 1, the course server plays some important roles in the whole integrated system. On the development side we anticipate agent technology will be use to implement courses comprised of learning objects (Lin, Holt, Korba, and Shih, 2001)[7]. To enable computer agents to automatically and dynamically compose personalized course materials for an individual learner, we need some effective knowledge management mechanisms and to include instructional design information in the metadata. Presumably learning objects may include their own agents and our system agents would have to negotiate with these "foreign agents. For delivery it must be able to store a variety of learning objects consisting of text, image, graphs, audio and video clips, and use them for composing courses. The course server serves both general web browsing clients such as Netscape and Internet Explorer, and our specialized system. The server has a JASIG u-portal (http://www.ja-sig.org/), which will include standard channels for online testing, external links, agent enhanced conferencing, agent dispatch, and student white pages (and yellow pages) for peer-to-peer networking. An agent maintains the link database removing broken links and notifying students' agents of links of particular interest to their owner. Some of these services will be presented in the browser; some may interact directly with agents on the student's computer. Some special attention must be paid to course server support for data encryption, user authentication and some more complex security operations, as well as support for the implementation and deployment of intelligent mobile agents within the system.

A course delivery module on the server is designed to link course servers to course clients running on student's computers. This module plays four important roles. The first role is to generate individualized course contents for each individual student based on the course material and students' information in the student database. Some course materials such as assignments, external links and other dynamic course material may be kept on the central course server. The second role is to serve the centrally delivered materials. Other materials are downloaded to ICSSL. The third role is to deliver the individualized course materials to the students. The fourth role is to manage the updating of downloaded materials). To reduce dependence on a live connection and enhance mobility the bulk of the course materials are distributed to the student's machine along with a student-learning module (SLM). A special mobile agent initially resides at the educational institution free the institution. It watches for connections from students. When there has been an update of the course materials since that last connection from a student it then makes a copy of itself, and travels to the student's computer with updating files specific for the identified student. It frees the learner from paying attention to

and spending time on updating and maintaining course materials.

Distributed Sub-systems

There are three distributed sub-systems: the sub-system for course coordinators (SCC), the sub-system for course tutors (SCT), and the sub-system for student learning (SSL). These systems share some modules such as the course messaging module, the course conferencing module, and the shared workspace module. Then the SCC has a module to help coordinators make decisions on student's evaluation and other academic-related matters involved in student management. The SCT has a module to help manage the duties of answering student questions in a timely fashion and demonstrating course contents to students. It includes a scheduler built in to help tutors to schedule their work. Much of the rest of this paper will deal with exploring the SSL as it is a relatively new and original concept for web-based learning environments.

The SSL (see Figure 2) consists of the learning management module, the course-messaging module, the course conferencing module, and the shared workspace module. The learning management module is designed to help students download and digest course materials, and access external documents. Standards-based educational material can be processed in a variety of ways according to educational features and passed to an appropriate viewer (for instance the XML enabled mozzilla embeddable browser, a multimedia viewer, or a simulations player). The browser will support shared browsing of course materials between two or more students (or a tutor and one or more students). There are many multimedia-viewing programs but they do not include many of the functions useful for learning such as tools for testing the recognition of specific situations in a video clip. This module includes an interface to the many tools a learner might use for his course. Finally, since it supports many functions at a local level, ISSL allows mobile users to accomplish their learning on the move. With the course messaging module students can communicate with course coordinators and tutors and submit their assignments directly without having to switch to other mail readers. Instant messaging, chat, audiochat, and video chat will be optional sub-modules of this module. The course conferencing module provides a posting management system for course conferencing. Finally the workspace-sharing module allows students to share files during collaborative work. Overall the SSL will be an ideal learning environment for collaborative work and will support professional practices such as extreme programming.

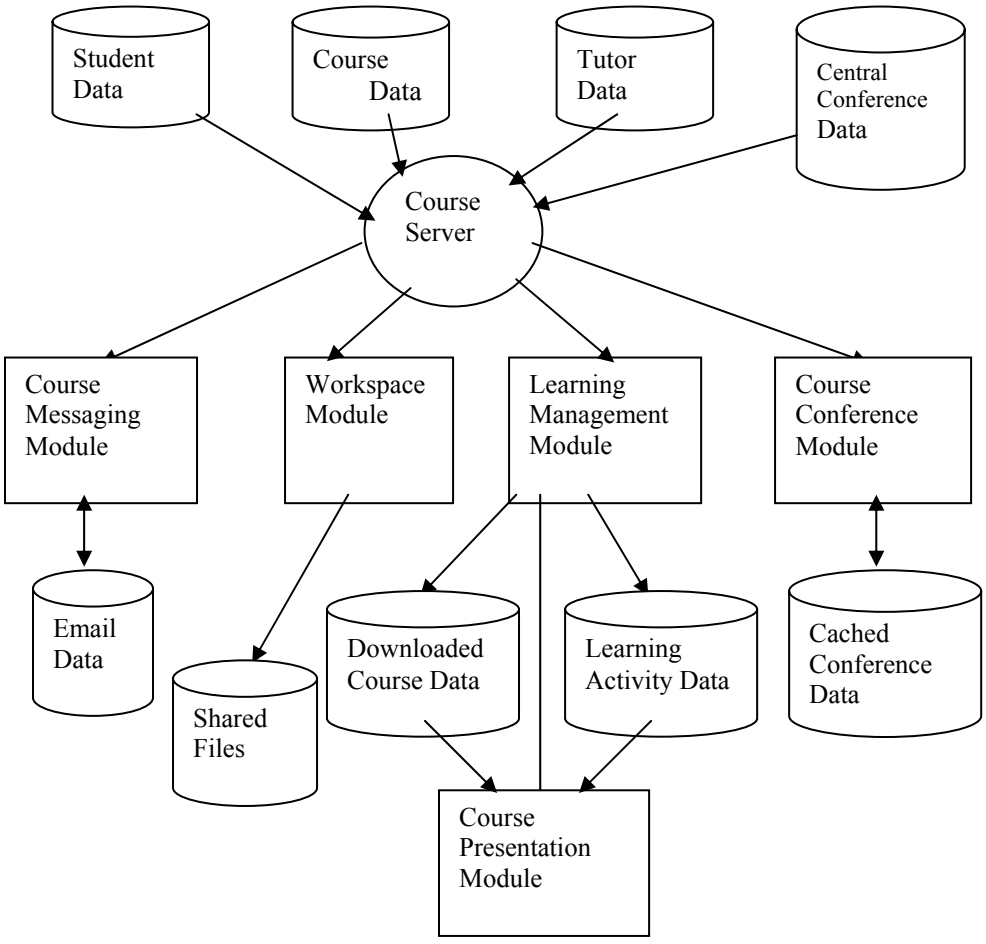


Figure 2 Sub-System for Student Learning (SSL)

Lin and Holt, 2001[8], have outlined how distributed intelligent agents can be incorporated to enhance the functionality of DALE student profiles (for student modeling) and access to course materials. Student agents reside primarily on the student client after being dispatched from an agent archive on the server. Functions supported by agents include collaborating with other students, generating self-tests of random items, tailoring course material to student profile,

managing course material presentation, managing external links (reporting new links to of interest to the students). There is no reason to restrict connections to a single institution. Learner's agents could reside on the learner platform interacting with a variety of institutions and peers from those institutions. Some agents may be mobile and visit a variety of hosting institutions looking for particular courses, information, or peer support.

4. IMPLEMENTATION STRATEGIES

CCIS uses open source and free software following open systems standards so we are not locked into a particular vendor's solution. This approach providea e-learning environments that are not under the control of commercial interests. Also it provides CCIS with an appropriate research focus for a computer science program at a distance education university. However, it is clear that neither a single department nor even a single university will be able to develop, implement, and support a complete open learning environment. The other options are commercial proprietary closed systems, proprietary open source systems, or open source systems developed by formal and informal collaborations of educational institutions and government. Hence, CCIS is involved in or is tracking a number of open source and related collaborative endeavors: JA-SIG (http://www.ja-sig.org/) is an independent organization whose mission is to enhance the flow of information among educational institutions and companies involved in the development of administrative applications using Java technology. The JA-SIG uPortal is a free, sharable portal for post-secondary institutions. It is an open-standard effort using Java, XML, JSP and J2EE and a collaborative development project. The Open Knowledge Initiative (http://web.mit.edu/oki/) led by the Massachusetts Institute of Technology and Stanford University is developing open source software for e-learning. The software is an alternative for institutions that want to provide online courses but do not want to invest in commercial course management systems. The mozilla project (http://www.mozilla.org/) is developing an embeddable xml-enable browser that can be embedded with a Java application (http://www.mozilla.org/projects/blackwood/webclient/) as part the learner's platform in a distributed learning environment. SUN supported Project JXTA (http://www.jxta.org/) is an open source effort aimed at providing "an open, generalized protocol that interoperates with any peer on the network including PCs, servers and other connected devices" to facilitate the development of distributed applications. GNOME (http://www.gnome.org/) is an open source desktop environment for Linux..
With the emergence of XML with DTD's and schemas, X-Path, X-link, XML objects (DOM) and XSL, tools are emerging for building inheritance hierarchies for all curriculum objects that can be easily manipulated by Java. JMF allows us to easily integrate and manipulate multimedia. For foreseeable future we will continue to build on XML, Java, and Java related tools For example for agent communications we will use the Java Reflection Broker (JRB: http://andromeda.cselt.it/users/g/grasso/free.htm) which deploys a dynamic invocation model similar to the dynamic invocation interface in the Common Object Request Broker Architecture (CORBA). We use a KQML-like message protocol and the Java Bean event model to provide the communication mechanism between agents and the facilitator and define our own event objects to handle the message content. We are considering using SOAP (http://www.w3.org/TR/SOAP/) for XML document exchange. We add security using the Java Security API, including user authentication and data encryption. Our peer-to-peer technology will be based on JXTA (http://www.jxta.org).

5. CONCLUSION

Based on agent-based systems, XML, and learning objects technologies, DALE will be very important to provide a body of electronic instructional support tools, curriculum content, and design strategies from which, on the basis of learners needs, they can adaptively select materials for a particular course. This also builds a basis to join the learning objects marketplace outside AU.

Distance Education can be considered as a service. A better service will attract more students. As an example of AU, DALE will be important to provide better learning services to students by using intelligent software agents (and Push technology) which will be able to help students select courses, seek for peer helpers or learning communities, notify web course material changes, provide useful web links, learn from multiple collaborating tutors with their agents, etc.

DALE addresses freedom of information and privacy issues, and scalability issues in novel and more effective ways due to its distributed nature. DALE will consider learning environment at an infrastructure level (instead of single PC and network level) of distance learning, considering student mobility and interface variety, which providing great convenience and flexibility to students.

REFERENCES

[1] Race, Phil. *The Open Learning Handbook*. NJ: Nichols Publishing Company.Review of Design, Teaching, and Institutional Issues. PA: ACSDE. 1994.
[2] Holt, P., Fontaine, C., Gismondi, J. and Ramsden, D. Collaborative Learning Using Guided Discovery on the INTERNET. *ICCE95*. Singapore. 1995.
[3] Lin, F., Holt, P. Towards Agent-based Online Learning, *CATE'2001*, Banff, Canada. 2001.
[4] Gelernter, D. The second coming a manifesto. Edge, 70, June 15. 2000. URL http://www.edge.org/documents/archive/edge70.html
[5] Oram, A. (Ed.) Peer-to-Peer Harnessing the Power of Disruptive Technologies, O'Reilly Sebastapol, CA. 2001.
[6] DALE Research Team. The Design of an Integrated System for Web-based Distance Education, CCIS Report in preparation. 2002.
[7] Lin, F., Holt, P., Korba, L. and Shih, T. A Framework for Developing Online Learning Systems, *DMS'2001*, Taipei, Tiawan. 2001.
[8] Lin, F., Holt, P. Towards Agent-based Online Learning, *CATE'2001*, Banff, Canada. 2001.

Virtual and Dynamic Hierarchical Architecture FOR e-Science and Related Protocols*

Huang Lican, Wu Zhaohui, Pan Yunhe
College of Computer Science and Technology
Zhejiang University
Hangzhou, Zhejiang , China, Post Code 310027
E-mail: lchuang@cs.zju.edu.cn

ABSTRACT

E-Science is commonly related to Grid in the domain of science and engineering. The architecture of e-Science is an important aspect in Grid architecture. However, there is no consensus about the architecture topology and the way of the information exchange among grid nodes yet. This paper presents an e-Science architecture known as Virtual and Dynamic Hierarchical Architecture (VDHA). VDHA is suitable for autonomous systems such as Internet; VDHA makes it feasible to manage users' privileges and roles; VDHA has high performance to discover services; Most of the messages in VDHA only relate to the neighboring two-layers, not the entire grid network. In this paper, the advantages and several protocols of VDHA are discussed.

Keywords: VDHA, Grid, e-Science, protocol

1. INTRODUCTION

"e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it."[1]. e-Science enables scientists to generate, analyze, share and discuss their insights, experiments and results in a more effective manner. The main characteristics of e-Science are coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations called as VO[2], and dynamically involving a large number of nodes generally distributed globally in geography.

The computing architecture of e-Science is usually based on Grid [2]. The newly proposed Open Grid Services Architecture (OGSA) [3] of the grid integrates the so called computational/data grid architecture [2] with Web services [4], and this architecture is a service-oriented architecture. From data exchange point of view, the architecture of e-Science can be classified from raw data to knowledge data as three-layered services architecture: Computation/Data service Grid, information service Grid and knowledge service Grid [5].

There are two major types of computing models. The prevalent client-server model is suitable for slim hosts as clients, especially mobile apparatus such as palm computers, but it may cause a performance bottleneck and an entire breakdown due to a single point of failure. Peer to Peer (P2P) model [6] can solve the scalable and fault tolerance problems, but it has some challenges such as security, network bandwidth, and architecture designs. We present VDHA to combine the advantages of the above two models and avoid the shortcomings in such a way that all the nodes may dynamically join and leave the virtually hierarchical group, and that the clients only attach one of the nodes.

VDHA is suitable for autonomous systems such as Internet, which are prerequisite for scalability; VDHA makes it feasible to manage users' privileges and roles; VDHA has high performance to discover services; Most of the messages in VDHA only relate to the neighboring two-layers, not the entire grid network. In this paper, we describe VDHA, its advantages, and several protocols related to this architecture.

The structure of this paper is as following: Section 2 describes VDHA, Section 3 presents related protocols, Section 4 gives an example about virtual cooperative research projects granted by China Educational Ministry, and finally we give discussions and conclusions.

2. OVERVIEW OF VDHA

Virtual and Dynamic Hierarchical Architecture (VDHA) (see Fig.1) is a dynamic and virtual hierarchical architecture to overcome shortcomings of P2P and to fulfill the requirements of virtual organizers. Client ends are attached to one of the nodes which are grouped virtually. Client ends and their attached nodes use Client/Server Mode. The nodes in the same virtual group use P2P architecture. Nodes can join the group and leave the group dynamically. The groups are virtually hierarchical, with one bottom-layer (these groups are called VOs), several up-layers, and one root-layer. From the nodes of VOs, one and only one node (called as gateway node) in each group is chosen to form second-layer groups, from the nodes of these second-layer groups to form upper-layer groups in the same way, and this way is repeated until to form one root-layer group. In the same group all nodes are capable to be gateway node. Gateway node is the node which is not only in low-layer group, but also in up-layer group. Gateway nodes will forward the low-layer group's status information to all the nodes in the up-layer group, and distribute the upper-layer group's status information to all the nodes in the lower-layer group.

Generally, the nodes in the same group have the same domain with similar properties. The nodes from within the same group exchange information more frequently than the nodes from within different groups. For the convenient management of authorization, it is reasonable to think that nodes in the same group have some common privileges and roles, and nodes can inherit privileges from the ones of upper-layer group. One node can join more than one VO. The numbers of nodes in a VO can be dynamically changed by the way that the node can dynamically join and leave the VO. A VO may join and leave the Grid system as a whole, and this autonomous property makes the large scalable systems possible.

*This paper is supported by Virtual cooperative research project granted by the Ministry of Education of PRC.

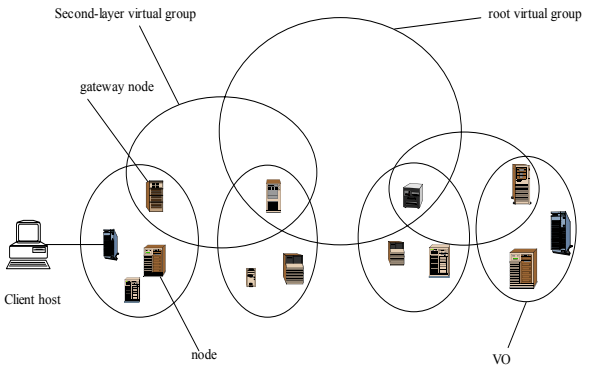


Fig.1 Structure of VDHA

Note: there are 16 nodes in the grid system. These nodes are grouped as 4 VOs. The number of nodes in each VO is 4,3,3,3 respectively. From each VO we choose one node as gateway node to form two up-layer groups with each having 2 nodes. Then from these two groups, one node each was chosen to form a root group.

3. SOME RELATED PROTOCOLS of VDHA

e-Science has many requirements such as security, authentication, scale, services discovery, performance, etc. VDHA uses the simple strategy to fulfill these requirements. The gateway nodes are used to exchange messages between two neighboring layer groups, rather than every node exchanges message with each other. Status information is generally only distributed within two neighboring layer groups, not entire network. All the above strategies extremely simplify the Grid implementation and make scalability of a large number of nodes possible. Several protocols related to the above strategies of VDHA are described as follows:

Gateway Node Selective Protocol (GNSP)

Gateway Node Selective Protocol (GNSP) is an important protocol for selecting gateway node, because gateway node is the key node in VDHA. The gateway node choosing algorithm is as following:

while(True) {
  switch(event) {
    case : a VO wants to join e-Science grid system: The node which has more resources is chosen as gateway node to join an up-layer group.
    case : a VO wants to leave e-Science grid system
      If the gateway node of this VO also is the gateway node in upper-layer groups, the gateway node in upper-layer is replaced with the on-line node with maximum weight value, otherwise, only remove these gateway nodes.
      case : a gateway node want to leave a VO
      The gateway node is replaced with the on-line node with maximum weight value except of this leaving gateway node.
      case: the gateway node fails to transmit messages between two neighboring layers exceeding a given times
      The on-line nodes with maximum weight value except of this failed gateway node is chosen as gateway node.
      case: one of the above 4 cases
      Update the nodes' state data (node name, weight value, etc. in the two neighboring layer.
  }
}

The weight is generally decided according to node's resources with the global coordinate specification by manual. The gateway node is chosen by the way that the node with the maximum weight is first chosen. If the weight values of several nodes are the same, then random node is chosen.

Grid Group Management Protocol (GGMP)

Grid Group Management Protocol (GGMP) is the protocol for virtual group management which is similar to IGMP [7].

There are several situations to group management as following:

1. If a node wants to join or leave a VO, then it forwards its requisition to a neighboring node. If the neighboring node accepts the node requisition, it forwards the joined/left node's information to gateway node, and the gateway node forwards this node information to all the nodes in the two neighboring layer groups.
2. If there are any changes in any nodes, then they are forwarded to gateway node. The gateway node forwards this change information to all the nodes in the two neighboring layer groups.

Query and Discovery Protocol (QDP)

Query and Discovery Protocol (QDP) is used for querying and discovering some entities such as resources and services. Every node has resources and services which are described by resource and service description language such as MDS [8] and WSDL [9, 10]. If one node wants to find a node which can support a given resource or service, it forwards this query to a gateway node, and this gateway node forwards this query message to all of the nodes in the same group in which the asking node is within. If the resource or service is found, then the querying is the end. Otherwise, the gateway node forwards the query message to all nodes in the up-layer group, and all the gateway nodes in the up-layer group forward this message to their lower-layer nodes to query the resource or service, and so on. (If we want to query the best service in the entire network, then we do not stop to search if we find a service, instead we search all the nodes in this way to find the best service.) (See Fig. 2)

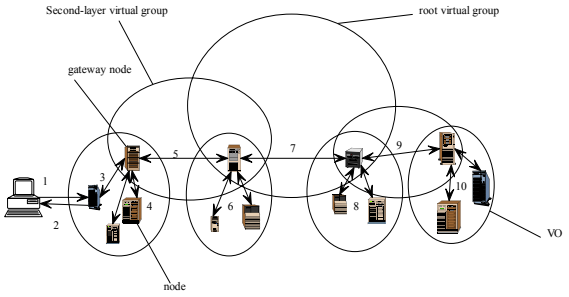


Fig.2 Scenery diagram of query one service in Query and Discovery Protocol (QDP)

Note: the meanings of the steps are as following:

1. Client host sends service query to entrance node.
3. The entrance node forwards the query message to gateway node
4. The gateway forwards query to all other nodes in the lower-layer group. If services are found, return the answer message until to client host and stop.
5. The gateway node forwards query to all other gateway nodes in the up-layer group.
6. The other gateway nodes forward query to all other nodes in the lower-layer group. If services found, return back the answer message to client host and stop.
7. The second-layer gateway nodes forward query to other

gateway nodes in the third-layer group.
8. The nodes in the third-layer group forward query to other gateway nodes in the second-layer group.
9. The second gateway nodes forward query to other nodes. If services found, return the answer message until to client host and stop.

This protocol has the following advantages:

- Firstly, the client does not generally know which node has the service. If the client queries every node, it will take much time, and spend much bandwidth. This hierarchical query method solves the above problems.
- Secondly, the client does not know the IP of every node because the e-Science grid system is dynamical, and with a huge numbers of nodes. This problem can be solved by QDP because in VDHA gateway node knows the information of every node in the two neighboring layer groups, and the numbers of nodes in these groups are small.

Security Architecture and Authentication Protocol (AP)
The security and authentication in VDHA are based on public key infrastructure. In VDHA, the domain node takes as CA of the users and generates the user's public key and private key. The domain node keeps its owned users' public key, and also some information of the owned users such as password, etc., which are used to identify user in ordinary ways. So this authentication policy is compatible with the common authentication policy used before joining Grid system. Because the numbers of nodes are smaller than the numbers of users, and for security and easy implementation reason, all the nodes' public keys are authenticated by CA center. This AP protocol is somewhat different with Globus GSI [11].

We use authentication ticket to solve the problems such as single-sign-on. Meanwhile, because the client host's IP address is generally LAN IP address, not the Internet IP address, we use the entrance nodes as proxy stations to help the client to connect to the Grid system. There are four modes about user's login. (See Fig.3)

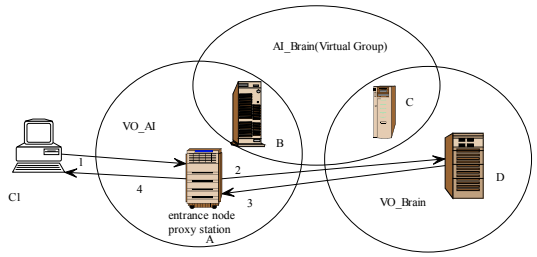


Fig.3 User login and authentication

(1).User logins remote owner node by using a client host
C1 is a client host, and the user is the account of node D in VO-Brain. The user via C1, which is attaching node A in the domain of VO_AI, logins to e-Science Grid system with remote mode. Login string is the string containing owner node's IP address, user name and password, for example, VO_Brain.D_IPaddr.lchuang, pwd= 123456. In Fig.3, The meanings of the steps are as following:
1. C1 generates the login request message and forwards this message to A node.
The login request message format is as follows:
owner-node-IP address
login-request
Login-request is encrypted with private key of user first and then user's owner node public key, e.g. encrypted with lchuang's private key first, and then VO_Brain.D node's public key. Login-request contains authentication ticket and user's Password in its owner node.
Authentication ticket contains following items:
User-client-IP
User-entrance-IP
User-owner-node-IP
User-Account
Start-time
End-Time
Group-account
Group-account has the format like as supG1.
supG2.supG3.group.
2. Node A forwards this message to node D
3. Node D uses its private key and user's public key to un-encrypt the message part encrypted by the user, then check the user account and password. If all right, then Node D encrypts authentication ticket already encrypted by user's private key with the node D's private key again. Node D also encrypts receipted or rejected status by its private key. Then, Node D forwards this messages to node A. Node A un-encrypts the message using Node D's public keys. If in the message permission to login is indicated, then node A lets user in C1 login in, otherwise rejects the user's login.
4. Node A forwards the result of receipted or rejected status message to C1.

(2) User logins local owner node by using a client host
C1 is a client host, and the user is the account of node A in VO_AI. The user via C1 which is attaching node A of VO_AI logins to e-Science Grid system with local mode. In Fig.3, the meanings of the steps in this mode are as following:
1. this step is as the way as the step 1 of section of ".User logins remote owner node by using a client host"
2. no this step
3. no this step
4. Node A uses its private key and user's public key to un-encrypt the message part encrypted by the user, then check the user account and password. If all right, then Node A encrypts authentication ticket already encrypted by user's private key with the node A's private key again, and lets C1 login in. Node A forwards the result of receipted or rejected status message to C1.

(3) User logins remote owner node by using a node host
User is the account of node D in VO_Brain. The user by using node A of VO_AI logins to e-Science Grid system with remote mode. In Fig.3, the meanings of the steps in this mode are as following:
1. No this step
2. Request message is generated as the way as step 1 of section ".User logins remote owner node by using a client host". This login request message is forwarded to node D.
3. Node D uses its private key and user's public key to un-encrypt the message part encrypted by the user, then check the user account and password. If all right, then Node D encrypts authentication ticket already encrypted by user's private key with the node D's private key again. Node D also encrypts receipted or rejected status by its private key. Then, Node D forwards this message to node A. Node A un-encrypts the message using Node D's public keys. If the message permission is indicated, then Node A lets user login in, otherwise, rejects user's login.
4. No this step

(4) User logins local owner node by using a node host
User uses node A to login e-Science Grid system, and login is a local login. Request message is generated as the way as step 1 of section ".User logins remote owner node by using a client host". Node A uses its private key and user's public key to un-encrypt the message part encrypted by the user, then check the user account and password. If all right, then Node A encrypts authentication ticket already encrypted by user's private key with the node A's private key again, and let user login in. The result of receipted or rejected status is notified to the user.
For all of these four situations, the user's authentication is identified by authentication ticket. Because authentication ticket is encrypted by the user's private key and the private key of the user owner node, and authentication ticket is issued by the user owner node, authentication ticket can not be forged. So, every other node can identify the demander's identification by un-encrypting the received authentication ticket using the public keys of user and user's owner node. The public key of the user's owner node can be found in CA center, and the public key of the user can be obtained by asking the user's owner node.

4. A CASE STUDY

The virtual research projects granted by Chinese Educational Ministry aim to enhance the science and technology research by virtual cooperation via Internet. There are now 18 virtual organizers, each has a special domain. Each virtual organizer has average 6 nodes which are located in Universities or research institutes. For combining these 18 organizers into an e-Science Grid system, we use VDHA to model this e-Science Grid system prototype (called as Chinese University e-Science Grid CUEG). Eighteen nodes chosen from every 18 VOs each plus one Chinese Educational Ministry node form an up-layer virtual group. Initially, the nodes located in the primary institutes of 18 VOs are chosen as gateway notes.

In this prototype, some information must synchronize among all nodes between two neighboring layers. This information includes group names, every node's IP Address, weight value, current gateway node's IP Address, and so on. User can use portable computers as client hosts to remotely login into CUEG. The authentication has single sign on property. A node to join a VO needs to propose a join-query message to neighboring node. This join-query message includes node name, VO name, node IP, neighboring IP, weight value which is assigned by VO administrator manually (see Figure 4). When the neighboring node accepts the node as a VO member, the node's information will transfer to all the nodes of two neighboring layers, and the all other nodes of two neighboring layers will transfer to this node.

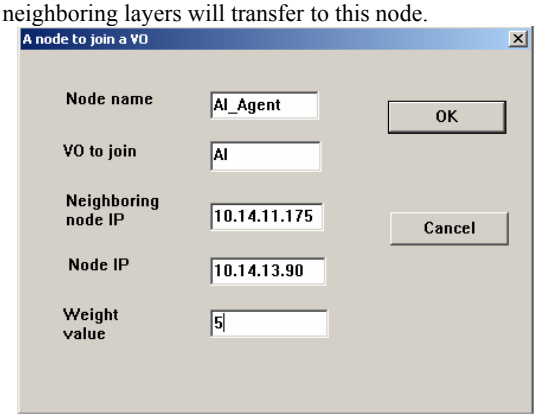


Fig.4 Dialog Window of a node to join a VO

In every CUEG node, there are some heterogeneous information depositories. How to share this information is an important issue. We implement a middleware retrieve service, which is modified from DHISWM[12]. The items of this service and metadata to describe the heterogeneous data are as follows:

Service description:
Service name,
Metadata file name,
Owner,
User permitted to access,
Instance already or not,
Start-time

In metadata file, there are defined metadata of heterogeneous data as following:

Data element {data-element-ID, local name, type}
Field {field-ID, local name, type}
Data set {data-set-ID, local name, number of data elements (n), data-element-ID1, data-element-ID2, data-element-ID3, , , data-element-IDn}

Table {table-ID, local name, number of fields (n), field-ID1, field-ID2, field-ID3, , , field-IDn}

If a user wants to retrieve heterogeneous information, he can login to CUEG, and sends the service query using QDP. If there is a node having this service, and the user is authenticated by the user's authentication ticket, and in the description of service the item "user permitted to access" indicates the user name or the user's supergroup name, then the service is instantiated, and the user retrieves the information from this node.

5. DISCUSSION AND CONCLUSION

VDHA can solve the scale and autonomy problems. Some nodes can form a VO, and this VO can join the e-Science Grid without centralized administrator, only determined by its neighbor node. In VDHA the messages are generally only concerned with the nodes of the two neighboring layers, not with entire grid network. So, the e-Science Grid with VDHA has the possibility to become a huge net.
VDHA combine the advantages of Client/server and Peer-to-peer models. Suppose that there 100,000 nodes, in P2P model, 100000x100000 synchronized information message are required to exchange among nodes. However, in VDHA, if we use four virtual layers, and suppose 1000 VO groups each has average 100 nodes, 100 second-layer groups each with 10 gateway nodes, 10 third-layer groups each with 10 gateway nodes, one root-layer group with 10 gateway nodes, only about 1000x100 synchronized information message are required to exchange because the synchronized information message is distributed by the gateway node.

VDHA has high performance to discover services. To search a service all over the above demonstrative grid system, the client host must request 100,000 times, this will take a lot of time and spend much bandwidth. However, VDHA uses parallel searching mode, and only about 1000 requests are required if we distribute the request message with multicast mode.

VDHA may easily manage privileges and roles of users. The users can be grouped, and the groups may be a member of a supergroup, and so on. The group can inherit privileges from supergroup. So, a user, who is the member of the group which

inherits from a supergroup, can access the resource, if the privilege of accessing this resource is assigned to the supergroup by the authorization policy. This strategy has advantage for simplify resource authorization policy if the grid net is huge.

We have proposed the security architecture and authentication for VDHA. The security architecture fulfills the requirements of Grid such as single sign on, protection of credentials, interoperability with local security solutions, and scalability.

Our further work focuses on completing and enriching services of CUEG prototype, and on increasing nodes of CUEG.

6. ACKNOWLEDGEMENTS

Thanks specially to our colleagues and graduate students in our Lab for their discussions, cooperation and contribution.

7. REFERENCES

[1] John Taylor, http://www.e-science.clrc.ac.uk
[2] Foster,I., Kesselman, C. and Tuecke, S. “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, International Journal of High Performance Computing Applications , 15(3). 200-222. 2001, http://www.globus.org/research/papers/anatomy.pdf
[3] Ian Foster, Carl Kesselman, Jeffrey M.Nick, Steven Tuecke. “The Physiology of the Grid An Open Grid Services Architecture for Distributed Systems Integration”', 2002.2.17 http://www.globus.org/research/papers/ogsa.pdf
[4] Grid Web Services Workshop. 2001, http://gridport.npaci.edu/workshop/webserv01/agenda.html
[5] David De Roure, Nicholas Jennings, Nigel Shadbolt. “Research Agenda for the Semantic Grid: A Future e-Science Infrastructure”,2001,9 http://www.semanticgrid.org/v1.9/semgrid.pdf
[6] D. Clark. “Face-to-Face with Peer-to-Peer Networking”, Computer, Vol. 34, No.1, January 2001, pp.18-21
[7] R.Fenner. “Internet Group Management Protocol, Version 2, RFC 2236”, Nov,1997, http://www.rfc-editor.org/rfc/rfc2236.txt
[8] I. Foster and C. Kesselman. “Globus: A Metacomputing Infrastructure Toolkit”, International Journal of Supercomputer Applications, 11(2): 115-128, 1997. [3].
[9] Christensen, E., Curbera, F., Meredith, G. and Weerawarana., S. “Web Services Description Language (WSDL)” 1.1. W3C, Note 15, 2001, http://www.w3.org/TR/wsdl.
[10] “UDDI: Universal Description, Discovery and Integration”, http://www.uddi.org/
[11] Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. “A Security Architecture for Computational Grids”, In ACM Conference on Computers and Security , 1998, 83-91.
[12] Huang Lican, Wu Zhaohui . “Distributed Heterogeneous Inspecting System and Its Implementation”, Lecture Notes in computer Science, Vol 2480, pp 370-380.

Agent Modeling Technology By User in Distributed Parallel System

Diao Cheng_Jia
Information and Technology College, Nankai University
Tianjin China, 300071
E-mail: Diaocj@office.nankai.edu.cn

ABSTRACT

Based on Agent technology, the distributed parallel pre-warning system is an effective method, through which Network security can be improved. Generally speaking, it is the supplier who provides Agent software, which can be used to realize some specific functions. Therefore, users cannot modify or define an Agent. In this paper, a new user-defined Agent modeling technology is advanced, through which users can construct the models according to their requirements so much so that Agent can provide proper services in the special condition to help users realize their purposes. The adoption of the technology will make it possible that the multiple Agent collaboration, the adaptiveness, the mobility and security in varied environments will be realized. The technology is designed by means of the distributed object-oriented technology and component technology. The architecture, the main function module and modeling method of the system will be advanced and analyzed in the paper.

Keywords: Distributed, parallel, collaboration, Intrusion detection, User- defined Agent.

1. INTRODUCTION

Recently, Internet has been developing in a great speed, through which people can enjoy lots of conveniences, obtain freedom and make unlimited fortunes. Network has reached every part of human society, such as, politics, military, economics, technology, physical education and culture. Apart from fields mentioned above, methods of living, entertainment and communication have also been affected by network. It can be declared that the time of information has come. In addition to material and energy, information has become the third resources, without which human society can't exist. Furthermore, it will become the medium in the future. However, it should not be ignored that with the development of information technology, information revolution has begun and the national security has come into existence. As network has been developing towards larger scales and more openness, the information security of the network becomes more and more important so that more and more countries have paid attention to network security. Those who have controlled the network will control the world. Network security will cover fields, as follow: dangerous E-mail, threat to security from the virtual world, and the asymmetrical invisible intrusion – network attack. Lots of facts show that: information system and network are facing the great threats, which will bring about serious outcome and make large political and economical losses, so much so that the national benefits and national economy will be damaged greatly. Therefore, Internet security has become the important factor, which will have an effect on the further development of information society.

Internet has been developing rapidly such that network resources are organized in the scattered and distributed architecture, in which one LAN is connected to the other LAN through a gateway. In the same LAN, resource can be shared, and the gateway becomes the focus of security in the whole LAN. All the computers in the same LAN have the equal level of security. Therefore, the whole LAN will be broken, as long as the computer, acting as the gateway, is broken. Obviously, the distributed and parallel architecture should be adopted in the intrusion-detecting warning system of the LAN. The detecting software will monitor every server and computer. Although these scattered detecting software accomplish their tasks respectively, they must cooperate with each other to complete the task of intrusion detection.

At present, Agent technology has become the advanced method, through which the intrusion detection system of the large scale network can be realized effectively. There are many successful parallel distributed intrusion detection architectures, based on Agent, in the world such as:

1. Autonomous Agent for Intrusion Detection (AAFID): Agent clusters with the tree hierarchical architecture designed by Purdue University
2. Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD): The prototype system designed by Phillip Porras in SRI International, supporting the distributed agents in the parallel system.
3. Multiple hosts detection system characterized by the mobile agent technology, designed by IPA in Japan.

There are still many other systems, such as: Hummingbird designed by Idaho University, Java Agents for Meta_learning based on the intelligent agent and the Meta_learning technology, developed by Columbia University, and Intelligent Agents for Intrusion detection, based on the mobile intelligent agent technology designed by Lowa University etc.

Although the distributed intrusion detection can be used widely, there are still many things to be improved. There are many problems and difficulties when the technology is put into practice. These problems and difficulties comprise the research direction in the field of intrusion detection. In China, researches on this aspect just begin, and we are in the phase of tracing, material-collecting and introducing. After analyzing those models constructed in the world, we can see that these Agents are the software components designed by the developer in advance and can only be adjusted in the limited range, which should be permitted by the system. In other words, it can be completed by means of the adaptive ability and re-learning ability owned by the system, but the users can not change the functions of the components according to his or her needs. Therefore, it cannot support the access control mechanism, in which the different users can visit the system in the different modes. It is very hard for the designers to have the ability of foreseeing when they design the user interfaces. When system environment, functions or requirements change or can't keep accordance with the original design, the users can't modify and define an agent themselves. They have to ask the software developer to design the new components once again or replace the old components with the new one. Therefore, we advance the new technology, in which the users can define agent by themselves.

The distributed parallel intrusion detection warning system, based on the Agent technology, can be used as an effective

method, though which network security can be guaranteed. Usually, it is the software developer who produces Agent software for certain purposes, but the user can not modify of define an agent. In this paper, a new modeling technology is put forward, in which the users can define Agent according to their requirements. If the technology is adopted, the users can construct the model by themselves, such that Agents can provide proper services in certain cases to help the users realize their objects. In this technology, distributed parallel architecture, multiple Agents collaboration should be constructed. In addition, the adaptiveness, mobility and security need to be built up. The technology can be realized by means of the distributed object-oriented technology and component technology such that the system can be configured dynamically or cut to meet up all kinds of environment and requirements properly. From the discussion above, we can find that the system, constructed by using the agent technology, can be characterized by high expansibility and easy maintenance.

2. BASIC AGENT ARCHITECTURE IN THE DISTRIBUTED SYSTEM

Based on Agent technology, the distributed parallel intrusion detection system can be explained like this: it can be realized by running the software entity, which can monitor the status of security, Agent components. They can run automatically. In other words, they can only be controlled by the operating system rather than by other processes. There may be several Agents in a host. Based on Agent, the method can run continuously and have the ability to learn from experiences, apart from the ability to communicate and cooperate with other Agents. That is to say, Agent can be characterized by independence, self-adaptiveness, and self-learning. It also has the ability of transferring, information-collecting, and cooperating among nodes etc. The mobile agents will act like this: it transfers from one node to another; it can obtain information from the local node and also can configure the network detection dynamically by means of its mobility so that the attack can be traced dynamically. Every component is responsible for the communication among agents, receivers, monitors and users; there is a great deal of local monitors in a frame, which will provide the local result to the global detector array. On the other hand, the global detector array will confirm the warning. In the system, all Agents have the same modules, which can be showed in figure 1.

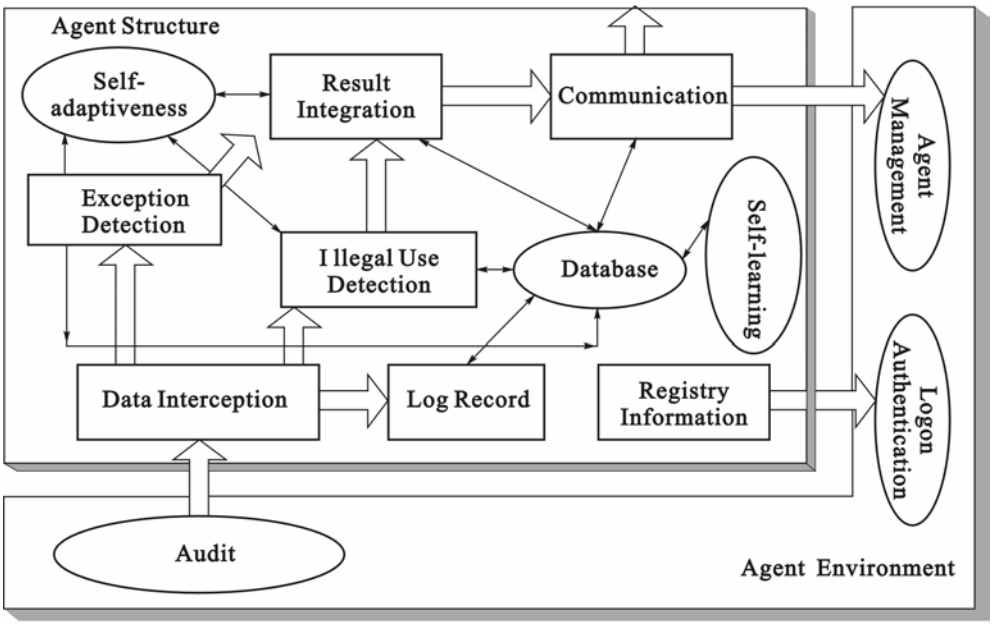


Figure.1 Agent structure and environment

Agent environment can provide the local support and authentication to Agent ; every Agent will obtain the needed data ,and system status etc from the environment. In order to support the mobile agent, log-on authentication and Agent management system will be added into the agent environment. Every Agent may have an information registration through the environment, informing that it has existed and will obtain a global identity number, by which communication with other agents can be conducted.

These Agents are the software components designed by the developer in advance, and can only be adjusted in the range permitted by the system. That is to say, it is completed by means of the self-adaptiveness and self-learning, but the access control mechanism will not be supported, in which the different users can visit the system in the different modes. It is very hard for the designer to have the ability of foreseeing when he or she is designing the user interfaces. When the great change has to be made, the new components have to be designed. In this case, it will not be convenient and also will take lots of money. Therefore, the new modeling technology by user through Agent is advanced.

3. INTRODUCTION TO THE USER-DEFINED AGENT SYSTEM

The so-called user-defined Agent modeling technology can be explained like this: through this technology, the user can build

up the model according to his or her needs, such that the agent can provide the proper services in the certain conditions in order to help the user reach the goal effectively. In this technology, the distributed parallel work mode can be constructed. It will also support the multiple Agent collaboration, self-adaptiveness, mobility and security in the varied environment. In order to realize the technology mentioned above, the object-oriented technology and component technology will be adopted.

Function Requirements

From the figure of Agent structure, we can see that the important components among Agent components will include: data interception, exception detection, illegal use detection, result integration, log record and communication etc. They will determine the functions of Agent. In other words, what to do and how to do it will be determined. Once they are determined, the structures of the basic database, the self-adaptive database and re-learning database, as well as the relative algorithm will be determined. At the same time, the range and requirements of auditing will be determined. As long as these function modules can be built up by the users, the functions of Agent will be determined by the users. And this will comprise the core of the user-defined Agent modeling technology.

Of course, from one side, one module is independent of the other, but from another side, they are relative. In a large scale LAN, every host may contain several agents. Although their architectures are identical, their functions are different. Some functions are complicated; some are simple. These software systems have to complete their own task, and also they have to cooperate with each other to accomplish the same mission. All these functions should be taken into consideration when the users begin to build their models.

Architecture

The system is made up of seven modules:

1) Module for audit: Defining the range and requirements of the auditing.
2) Module for the illegal use detection: Setting model of the illegal use , selecting methods for detecting, and selecting methods for integration.
3) Module for exception detection: Setting the model for exception detection, selecting the method for detecting, and selecting the method for integration.
4) Module for self-adaptiveness: Setting model and method for self-adaptiveness.
5) Module for knowledge-base of self-learning: Setting model and method for self-adaptiveness
6) Module for communication between agents: Defining communication between agents and communication between Agent and monitor.
7) Module for management of agents: Defining management methods of Agents, collaboration between agents and demands from one Agent to another.

Actually, some interfaces are set in modules mentioned above. All of these modules will be connected automatically according to the method and the structure selected by the user. Finally, the user-defined system will be created.

Modeling Methods

The user-defined Agent modeling method can be realized by using a verified instance, in which graphical hint and graph combination will be employed, in order that the newly created system by the user can be characterized by integrity, security and reliability. Certainly, the system must be supported in the platform, which should consist of a set of verified instances, models, data structures and a set of the relative algorithms.

It is these basic components that will make it possible for the users to define the agent model by themselves. The system created by the user can be put into a set of instances for future use, if the system proves to be reliable, secure and effective, after running for a period of time.

In the system, the user will be allowed to design the new models, the data structures and the relative algorithms. If it proves to be effective when being put into practice, it can also be added into the set so that it can be used as a component for future use.

4. CONCLUSION

The user-defined Agent modeling technology can be implemented by using the distributed object-oriented technology and the component technology. We have discussed the architecture of the system, the main function modules and the modeling methods. Of course, the final user of the system should be the system engineer, who is responsible for the maintenance of a LAN, and can learn how to use the system after a short period of training and learning. Eventually, they will be capable of designing the needed Agents through the system.

5. REFERENCES

[1] Wayne Jansen,Peter Mell,Tom Karygiannis et al. Applying Mobile Agents to Intrusion Detection and Response[M].NIST Interim Report (IR)-1999.10

[2] Porras P A, Neumann P G. EMERALD: event monitoring enabling responses to anomalous live disturbances. In: Proceedings of the 20th National Information Systems Security Conference. National Institute of Standards and Technology, 1997

[3] Rebecca Gurley Bace Intrusion Detection Macmillan Technical Publishing 2000

[4] Zhao Haibo et al, a real-time intelligent detection system for network intrusion. Academic Paper of Shanghai Jiaotong University 1992.2

[5] Zhang Yong et al, Researches and Implementations of Network Intrusion Detection based on the distributed, cooperation agents. Academic Paper of Computer vol.24, No7, July 2001

Research On Distributed Collaborative Optimization Technology *

Caijun Xue Qingying Qiu Peien Feng Jianwei Wu
State Key Laboratory of CAD&CG, Zhejiang University, HangZhou, 310027, P.R.China

ABSTRACT

Complex engineering optimization problems, such as design optimization of complex systems, are difficult to solve due to large-scale computation and complex simulation in order to compute a function value, so the distributed computing technology based on decomposition-coordination theory is very attention -getting to design engineers. Considering some disadvantages of the present computing methods and frames, this paper puts forward a distributed computing frame, Collaborative Subspace Optimization Frame (CSOF). Then, the paper delves into the coordination problem of distributed collaborative optimization systems for dealing with the conflict among related variables in sub-optimization problems. A complete coordination model is built, in which the coordination process is treated as an optimization problem itself. And a genetic algorithm based coordination strategy of sub -optimization is studied. The collaborative optimization system is implemented using multi-agent technology and the design and implementation of optimization agents are discussed in detail, and CORBA technology is used to implement communication between the components of the optimization agents. An example demonstrates the computing method efficient and the multi-agent system reliable and flexible.

Keywords: Engineering Optimization, Large-Scale Computation, Decomposition-Coordination Method, Collaborative Optimization, Coordination Strategy, Multi-Agent System.

1. INTRODUCTION

Since products development is an optimization process in nature, it is an important applications field of optimization technology. However, the present optimization techniques can't yet meet the demand of modern engineering design problems due to large-scale computation when we start considering design of the overall system and/or the overall design process. Therefore, it is a much focused research area to provide engineering support and solutions to design optimizations. Collaborative optimization is an approach by decomposing large design problems into smaller design problems, which can be assigned to different groups of engineers. Decomposing a large optimization problem into several smaller optimization sub-problems make it possible that many optimization problems are solved in parallel. Thus, efficiency is increased. The requirements of Multidisciplinary Design Optimization (MDO) and complex problems in product optimization design accelerate the research on collaborative optimization. A few valid approaches for collaborative optimization have been developed, such as constrained subspace optimization, Bi-level integrated system synthesis (BLISS) and collaborative optimization approach [1, 2, 3, and 4]. However, these approaches are mainly focused on integrating developed modules, lack of flexibility and can't support product development process because a necessary systemic optimization model is difficult to build and information of analytic gradients is usually difficult to obtain. The paper puts forward a distributed computing frame--Collaborative Subspace Optimization Frame (CSOF) and a distributed computing frame and a multi-agent based collaborative optimization system is constructed.

The paper is organized as follows: a distributed computing frame is presented and the advantages of it are summarized in section 2. The complete autonomy of sub-optimization problems makes the conflict among them more serious. So, in section 3 the coordination strategy among sub-optimization problems including establishing the coordination model and computing strategy of the coordination model, is delved into. In section 4, the field ontology of collaborative optimization is studied and the architecture of a multi-agent system is presented based on the field ontology. The section 5 will formulate the model of optimization agents and discuss the implementing method based on CORBA. A demonstrating example is provided in t section 6. Finally, conclusion will be drawn in section 7.

2. COLLABORATIVE SUBSPACE OPTIMIZATION FRAME

Products development is a collaborative process in which each design group deals with a sub-system. As a result, the sub-optimization models are independent a certain extent. At the other hand, the correlation of the sub-systems results in the coupling relationship of the sub-optimization models. Our aim of studying design optimizations technology is developing practical design optimization tools for engineers enabling them to systematically improve designs and manufacturing processes and deals with issues such as integration of optimization, and distributed computing into the overall design and manufacturing process. Therefore, It is clear that a collaborative optimization frame should meet the following demands:

[1] Avoiding modifying any sub-optimization models because modifying the optimization models not only demands design engineers to grasp collaborative optimization algorithm, but also may lead to loss of technology security.
[2] Sub-optimization can be computed independently in order to validate sub-optimization models.
[3] Sub-optimization program runs in parallel mode and the collaborative operation can be finished intelligently under the frame.

Based on the above considerations the paper puts forward a collaborative optimization frame shown by Fig.1, named Collaborative Subspace Optimization Frame (CSOF). Sub-optimizations run in the independent sub-spaces and the

* This research is supported by the National 863 Program of China under Grand Number No. 2001AA412110 and the National Science Foundation of China under Grant Number No. 59635150.

coordination module (Co. Opt.) will deal with conflict of sub-optimizations when the coordination condition is met. Sub-optimizations are more independent and easier to run in parallel under the frame and this frame is more flexible. Moreover, technology security of design groups is guaranteed since it is not necessary to modify sub-optimization models. Difficulty to apply the frame is to develop an efficient coordination strategy to solve the conflict among sub-optimization problems.

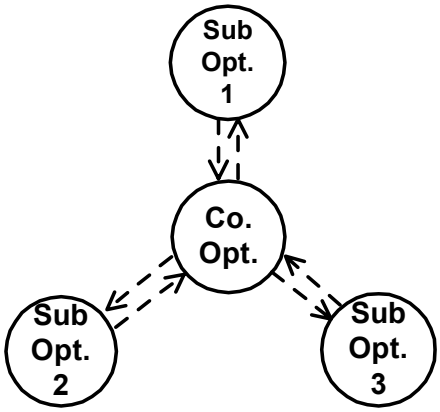


Fig. 1 Collaborative Subspace Optimization Frame (CSOF)

3. COORDINATION OF SUB-OPTIMIZATIONS

Problem description

The complete autonomy of sub-optimization problems makes the conflict among them more serious, which must be resolved by a coordination strategy. An optimization problem including three sub-optimizations is considered in the paper. The designs variables are classified into free variables and related variables. The coupling relation of the related variables is denoted in Fig 2.

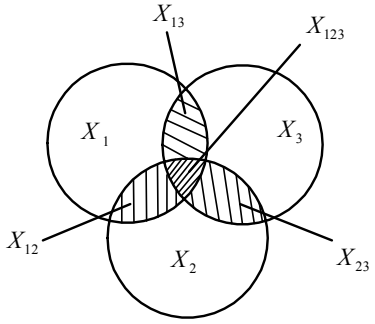


Fig 2 Coupling relation of the related variables

The model of sub-optimization is formulated as Eq.1:

$$Min \quad F_i(X_i)$$
$$s.t. \quad G_i(X_i) \le 0 \qquad (1)$$

The variables of a sub-optimization problem can be formulated as Eq.2, Eq.3 and Eq.4:

$$X_1 = \{X_{1f} \;\; X_{12} \;\; X_{13}\} \qquad (2)$$
$$X_2 = \{X_{2f} \;\; X_{12} \;\; X_{23}\} \qquad (3)$$
$$X_3 = \{X_{3f} \;\; X_{13} \;\; X_{23}\} \qquad (4)$$

Where, X_{1f}, X_{2f} and X_{3f} are free variable vectors, and X_{12}, X_{13} and X_{23} are related variable vectors. And, $X_{123} = X_{12} \cap X_{13} \cap X_{23}$.

Coordination model of sub-optimization

When a sub-optimization puts forward a coordination proposal, the related sub-optimizations stop optimization process, and then collaborative process starts. The present values of the related variables for every sub-optimization can be formulated as Eq.5, Eq.6 and Eq.7 :

$$X_1^0 = \{X_{1f} \;\; X_{12}^1 \;\; X_{13}^1\} \qquad (5)$$
$$X_2^0 = \{X_{2f} \;\; X_{12}^2 \;\; X_{23}^2\} \qquad (6)$$
$$X_3^0 = \{X_{3f} \;\; X_{13}^3 \;\; X_{23}^3\} \qquad (7)$$

The present values of the objective can be denoted as: $F_1(X_1^0)$、$F_2(X_2^0)$ and $F_3(X_3^0)$.

Since the purpose of coordination of the related variables is to find a variable scheme that satisfies every sub-optimization problem, the essential of coordination is an optimization problem as following:

$$Min \quad (F_1(X_1) - F_1(X_1^0))^2 + (F_2(X_2) - F_2(X_2^0))^2 + (F_3(X_3) - F_3(X_3^0))^2 \qquad (8)$$
$$S.t. \quad X_{12}^1 = X_{12}^2 \qquad (9)$$
$$X_{13}^1 = X_{13}^3 \qquad (10)$$
$$X_{23}^2 = X_{23}^3 \qquad (11)$$
$$G_i(X_i) \le 0 \qquad (i = 1,2,3) \qquad (12)$$

As a result of including all variables in the above model, objectives and restrictions of sub-optimization problems in the system, the model has unambiguous meaning and good integrality. However, it is nearly impossible to calculate directly the above model, and therefore studying simplified calculating method is obligatory.

Simplified calculating method

The simplified method simulates biology evolution process and by crossing and mutant of the related variables produce new design schemes. As a result of natural selection based on valuating the objective of collaborative model, descendants differ from their ancestors. Fig 3 shows the theory of this simplified method that includes three main steps.

First step is to form new design schemes by crossing and/or mutant of the related variables. The crossing operation of the related variables is defined as following:

$$X^I = X^1 \otimes X^2 \qquad (13)$$

Where, $\otimes$ is defined a crossing operator. If

$$X^1 = \{x_1^1 \; x_2^1 \; x_3^1 \; \cdots \; x_n^1\} \qquad (14)$$
$$X^2 = \{x_1^2 \; x_2^2 \; x_3^2 \; \cdots \; x_n^2\} \qquad (15)$$

The theory of crossing operation is demonstrated as following, and the crossing place can be selected at random.

X^1	x_1^1	x_2^1	x_3^1	$\cdots$	x_n^1
X^2	x_1^2	x_2^2	x_3^2	$\cdots$	x_n^2
X^I	x_1^1	x_2^1	x_3^2	$\cdots$	x_n^2

The mutant operation of the related variables is defined as following:

$$X^I = X^1 \odot X^2 \qquad (16)$$

Where, $\odot$ is defined a mutant operator. The theory of mutant operation is demonstrated as following, and the mutant place can be selected at random.

X^1	x_1^1	x_2^1	x_3^1	$\cdots$	x_n^1
X^2	x_1^2	x_2^2	x_3^2	$\cdots$	x_n^2
X^m	x_1^1	x_2^1	x_3^m	$\cdots$	x_n^1

Where, $x_3^m = (x_3^1 + x_3^2)/2$.

The second step is a local optimization process for each new scheme about the related variables coming from the first step, where the related variables don't participate in optimization. Every sub-optimization problem finishes the process to get local optimal solutions. Finally, the objective of coordination model is calculated, based on which the optimal solution is selected.

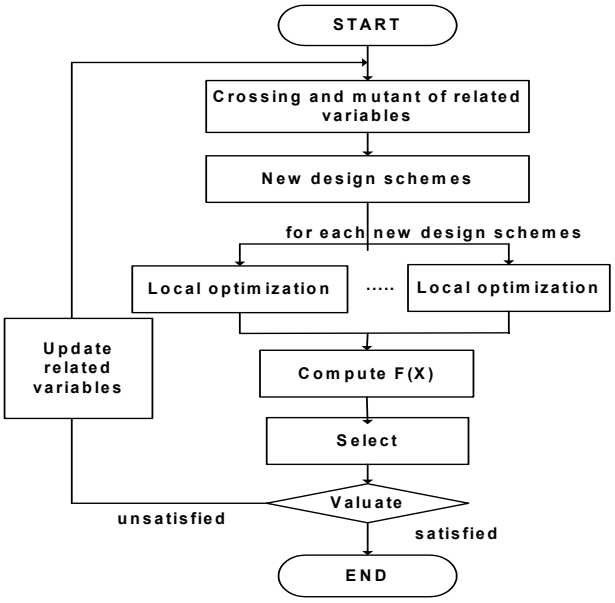


Fig 3 Calculating method of coordination model

The converging condition is defined as $\omega = (Obj_k - Obj_{k-1})/Obj_k$. When $\omega \le \omega_0$, the coordination process stops and the optimization agent restart based on the coordination solution. When $\omega > \omega_0$, the solution, which the collaborative objective is smaller, is selected to produce new design schemes by crossing and/or mutant. The above process is a loop process until the condition $\omega \le \omega_0$ turns true.

It is clear that all the sub-optimization problems get the shared valuable preference by sharing the preferable information by crossing or (and) mutant of related design variables. So, the above coordination process is in fact an evolution process to the satisfied solution for all sub-optimization problems. A new and better starting point is provided.

4. THE MULTI-AGENT BASED COLLABORATIVE OPTIMIZATION SYSTEM

Field ontology of collaborative optimization system

Collaborative optimization system gains srong power for it's internal parallelity, flexibility and robust based on the distributed technology. However, components on system node is absorbed in the task on the field of itself, and restricted by it's concept structure (ontology), which cause the difficulty on cooperation and knowledge share. Hence, the methodology of ontology is introduced to set up the conceptual schema of collaborative optimization.

In the area of artificial intelligence, ontology belongs to theory of content, studies on the object classification, object property and relationship among objects on special field knowledge, and provides terminology for description of field knowledge [5]. Based on ontology principle, collaborative optimization conceptual schema is build up. Abstract gradation structure method is introduced to abstract collaborative optimization to be objects. (Fig. 4) According to the class tree KIF based technology is introduced to describe the object, process and relationship.

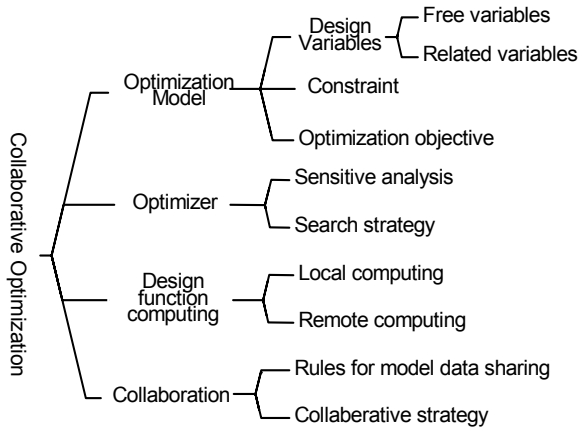


Fig. 4 Objective sort tree for collaborative optimization

Architecture of the multi-agent system

An agent is an entity with mental state such as belief, promise, obligation, intention and etc. Multi-agent technology desires the coodination in group with intelligence in some degree, and is an imitation to human society on rational level. The distributed architecture for collaborative optimization system is motivated by the following considerations: distributed architecture, autonomy, intelligence, robustness.

A multi-agent architecture for collaborative optimization is designed as Fig. 5 shows, including a management agent, a collaborative agent, a computing agent, several optimization agents and other utility agent (not shown in the figure). The management agent includes a user interface and a control module. The collaborative agent completes coordination among optimization agents. The computing agent responses to computing request (for example, FEA) coming from optimization agents, and allocate the computing tasks to computer in INTERNET according to the reliable computing resource. The whole system works together based on the field ontology.

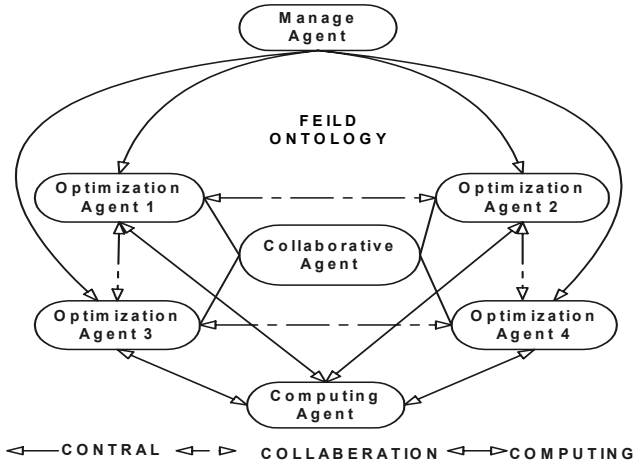


Fig.5 Architecture of multi-agent based collaborative optimization system

Running flowchart of the multi-agent system

The multi-agent based collaborative optimization system, written in the Java programming language, is running on the network and the running flowchart is demonstrated as Fig 6. First, the NameServer agent is started which facilitates information discovery. Then the user interface agent is started which allows user to control the while system. When the NameServer agent and user interface agent have been started, the remote optimization agents and computing agents can start and join the system. After all agents have started, the user can browse the related optimization models and computing resource based on which the collaborative optimization scheme is planed. Finally, the collaborative optimization process can be started by the user' direction.

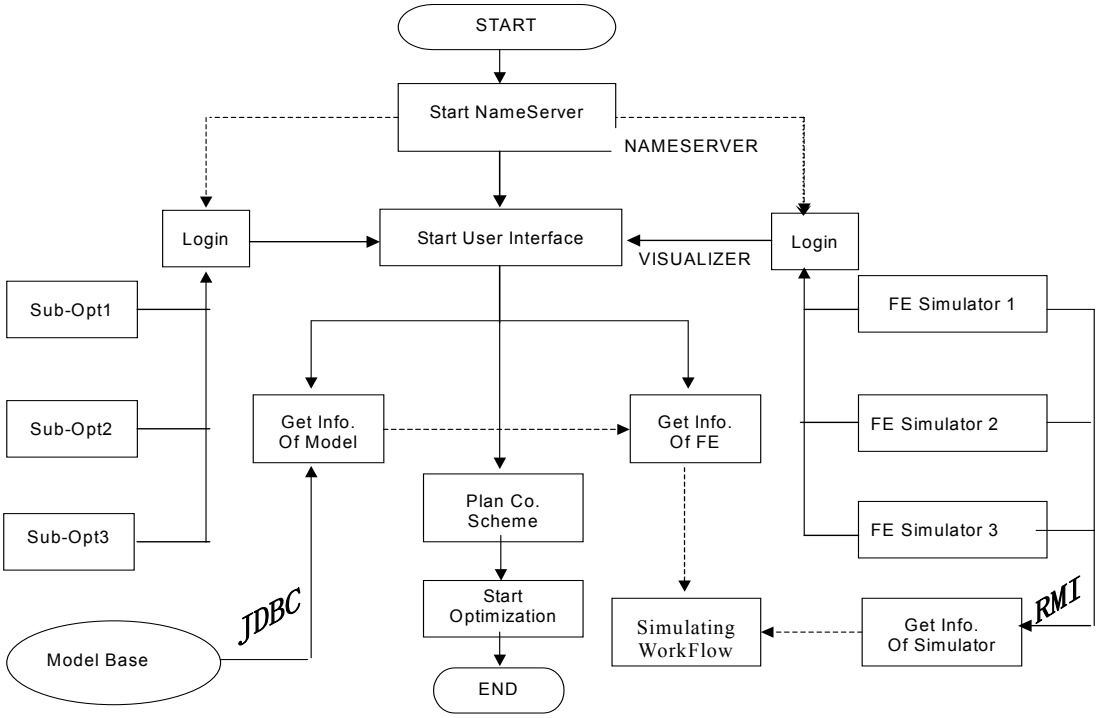


Fig.6 Running flowchart of the multi-agent system optimization system

5. OPTIMIZATION AGENT

An optimization agent should be an integration of an optimization program, an optimization model and an agent. Moreover, in order to increase effeminacy and flexibility, an optimization model and an optimization program should implement flexible integration based on the characters of the optimization model. Based on the se consideration an optimization agent is designed as Fig.7 shows.

Because the optimization program is usually written in FORTRAN language or C++ language and the agent is implemented in java language in our work, CORBA technology is used here to realize the communication between them as Fig.3 shows. CORBA is a transit structure, which can let multi-component communicate with each other on the network. Its characteristics include access transparency, network and location transparency, programming language independence, and hardware and operating system independence. The following interfaces are defined in an object of CORBA:

- ✧ AssignModel(…) Assign optimization model to optimizer.
- ✧ OptStart (…) Start optimization process.
- ✧ PauseOpt (…) Pause optimization process.
- ✧ StopOpt (…) Stop optimization process.
- ✧ GetVaribales (…) Get the values of variables.
- ✧ GetFunctions(…) Get the values of functions.

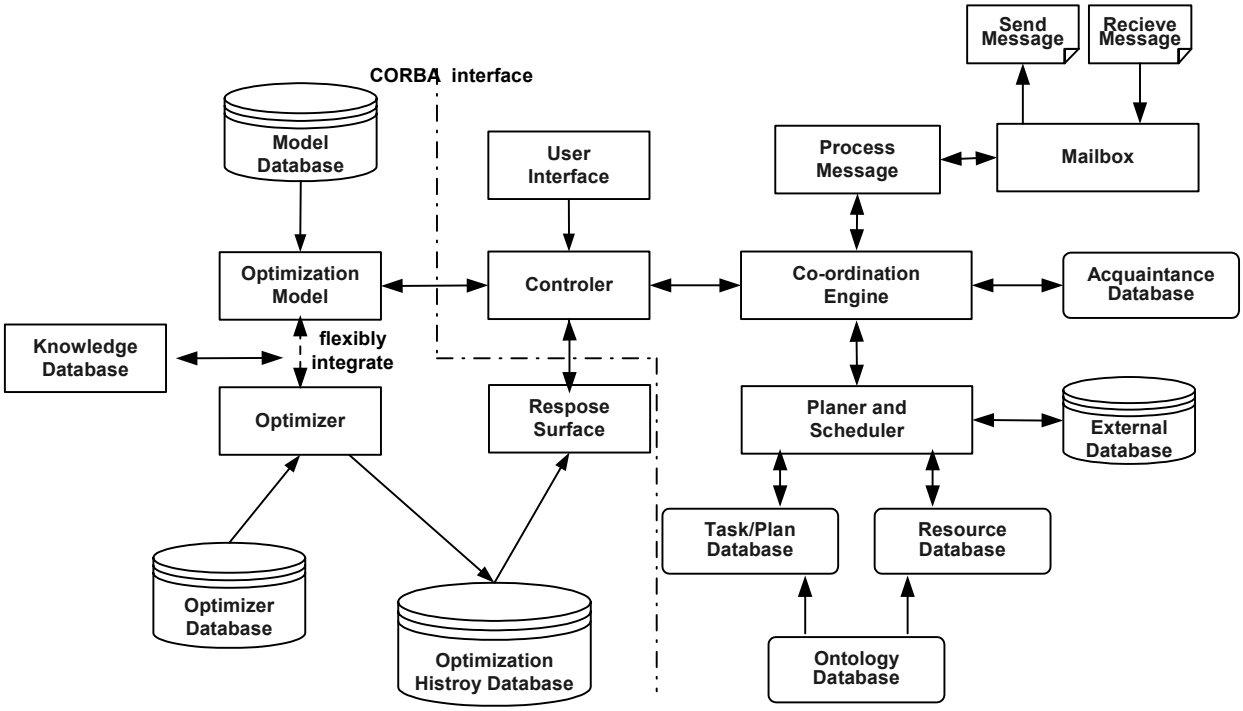


Fig.7 Model of optimization agent

6. A COMPUTING EXAMPLE

As an example, a gear reducer is considered, as shown in Fig 8. The design objective is to minimize the weight of the gear reducer under 11 constrains (see the sub-optimization models).

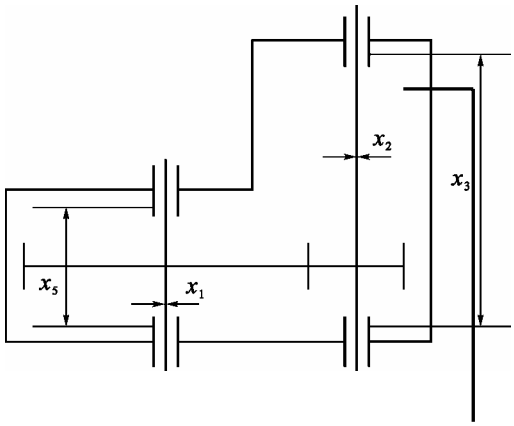


Fig 8 Optimal design of gear reducer

The meanings and the bounds of the design variables:
x_1: tooth width, $2.6 \le x_1 \le 3.6$,
x_2: module of gear, $0.7 \le x_2 \le 0.8$,
x_3: number of teeth of the smaller gear, $17 \le x_3 \le 28$,
x_4: distance of the first shaft bearing, $7.3 \le x_4 \le 8.3$,
x_5: distance of the second shaft bearing, $7.3 \le x_5 \le 8.3$,
x_6: diameter of the first shaft, $2.9 \le x_6 \le \le 3.9$,
x_7: diameter of the second shaft, $5.0 \le x_7 \le 5.5$.

The solution obtained by the whole optimization is listed as table 1.

Table 1 The solution obtained by the whole optimization

x_1	x_2	x_3	x_4	x_5	x_6	x_7	Obj. f
3.50	0.70	17.00	7.32	7.75	3.36	5.29	2997.9

The original problem is decomposed into three sub-problems as following:

Problem 1:

$$X^1 = \{ \; x_1 \quad x_2 \quad x_3 \; \}$$

$$Min \; f_1 = 0.7854 x_1 x_2^2 (3.333 x_3^2 + 14.933 x_3 - 43.0934)$$

$$S.t. \quad g_1 = 27/(x_1 x_2^2 x_3) - 1 \le 0$$

$$g_2 = 397.5/(x_1 x_2^2 x_3^2) - 1 \le 0$$

$$g_7 = x_2 x_3 / 4 - 1 \le 0$$

$$g_8 = x_1 / x_2 - 12 \le 0$$

$$g_9 = 5 - x_1 / x_2 \le 0$$

Problem 2:

$$X^2 = \{ \; x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_6 \}$$

$$Min \; f_2 = -1.508 x_1 x_6^2 + 7.477 x_6^3 + 0.7854 x_4 x_6^2$$

$$S.t. \quad g_3 = 1.93 x_4^3 / (x_2 x_3 x_6^4) - 1 \le 0$$

$$g_5 = \sqrt{745^2 x_4^2 / x_2^2 x_3^2 + 16.9 \times 10^6} \Big/ 110 x_6^3 - 1 \le 0$$

$$g_{10} = (1.5 x_6 + 1.9) / x_4 - 1 \le 0$$

Problem 3:

$$X^3 = \{ \; x_1 \quad x_2 \quad x_3 \quad x_5 \quad x_7 \}$$

$$Min \; f_3 = -1.508 x_1 x_7^3 + 7.477 x_7^3 + 0.7854 x_5 x_7^2$$

$$S.t. \quad g_4 = 1.93 x_5^3 / (x_2 x_3 x_7^4) - 1 \le 0$$

$$g_6 = \sqrt{745^2 x_5^2 / x_2^2 x_3^2 + 157.5 \times 10^6} \Big/ 85 x_7^3 - 1 \le 0$$

$$g_{11} = (1.1 x_7 + 1.9) / x_5 - 1 \le 0$$

The result got under the multi-agent based collaborative optimization system is listed in Table 2. From the table we can find that the result is very close to the result from the whole optimization. The objective obtained by collaborative optimization is listed in Table 3.

Table 2 Solution obtained by collaborative optimization

x_1	x_2	x_3	x_4	x_5	x_6	x_7
3.50	0.70	17.00	7.32	7.97	3.35	5.29

Table 3 Objective obtained by collaborative optimization

Obj. f_1	Obj. f_2	Obj. f_3	Obj. f
1580.35	286.28	1131.45	2998.08

7. CONCLUSION

The decomposition-coordination method has been used in complex engineering optimization for several decades. However, the successful application is very limited due to lack of flexible collaborative optimization system. This paper put forward a multi-agent based collaborative optimization system. In the multi-agent based collaborative optimization system, decomposing a large optimization problem into several smaller optimization problems make it possible many optimization agents working on the problems in parallel. The optimization of the working equipment of hydraulic excavator is implemented in this system and some useful results are obtained.

8. REFERENCES

[1] J.Sobieeszczanski-Sobieski, “Optimization by Decomposition: A Step from Hierarchic to Non-Hierarchic Systems,” NASA CP-3031.

[2] Srinivas Kodiyalam and Charles Yuan; Evaluation of Methods for Multidisciplinary Design Optimization (MDO), Part II, NASA/CR-2000-210313.

[3] Braun R., et al. Development and application of the collaborative optimization architecture in a multidisciplinary design environment. NASA, International Congress on Industrial and Applied Mathematics, Hamburg, Germany, 1995.

[4] Tappeta R.V., et al. Multiobjective collaborative optimization. ASME J. Mech. Des. 1997, 119(3): 403-11.

[5] Chandrasekaran B, Josephson J R, Benjamins V R. What are ontologies, and why do we need them? IEEE Intelligent Systems & Their Applications, 1999, 14(1): 20-26.

Robust Distributed System of Multi-Dimension Education Agents

Tao Gong and Zixing Cai, IEEE Senior Member
College of Information Science & Engineering, Central South University
Changsha, Hunan 410083, China
E-mail: eeslab@yahoo.com

ABSTRACT

In this paper, a novel Robust System of Multi-Dimension Education Agents (RSMDEA) has been proposed. RSMDEA is based on a distributed and parallel network-computing platform with multi-agent technique and distributed and parallel algorithms, such as co-evolutionary algorithms. RSMDEA can also be applied into multi-dimension education network platform or applications. Distributed and parallel architecture is the important feature of RSMDEA. Another important feature of RSMDEA is robustness. Based on problem reduction method, robust problem of RSMDEA in ideal distributed environment has been reduced and transformed into robust problems of single agents and discrete control processes. RSMDEA can be used widely in E-learning fields, especially for today's education network with too many unsafe challenges.

Keywords: Distributed Multi-agent System; Distributed and parallel algorithm; Model on reducing robust problem; Model on robustness relativity; Multi-dimension education system

1. INTRODUCTION

Distributed computing and parallel computing have been developed and applied widely in many fields, including business, engineering and science [1-5]. But function distributions according to actual requirement of systems and applications are often limited and stupid in traditional distributed systems. With the development of agent technology, agents like human beings can develop intelligence, adaptation, interactivity, and cooperation of distributed systems [6-8]. Besides, agent technology can expand the application fields and effects of distributed systems [9-10]. On the other hand, advanced performance analysis of multi-agent distributed system, such as robustness analysis, is still difficult for many researchers. Universal methods or reducing methods have not been proposed for this research.

In this paper, a novel Robust System of Multi-Dimension Education Agents (RSMDEA) has been proposed. RSMDEA is based on a distributed and parallel network-computing platform with multi-agent technique and distributed and parallel algorithms, such as co-evolutionary algorithms [11-12]. RSMDEA can also be applied into multi-dimension education network platform or applications [13-15]. Distributed and parallel architecture is an important feature of RSMDEA.

Another important feature of RSMDEA is robustness. Based on problem reduction method, robust problem of RSMDEA in ideal distributed environment has been reduced and transformed into robust problems of single agents and discrete control processes [16-18]. So problems have been simplified, and then research difficulty degree has been descended. Moreover by solving easier robust problems, robustness conditions of the whole multi-agent system can be found. Such has been called model and theorem on reducing robust problem of multi-agent system. For real distributed multi-agent environment, the model on robustness relativity of RSMDEA has been advanced, the relation between the ideal robustness reducing model and the real robustness relativity model has been analyzed, and robust simulation of distributed and parallel algorithms in RSMDEA can be created to test the models and theorems [19-20].

RSMDEA can be used widely in E-learning fields, especially for today's education network with too many unsafe challenges [13,16]. Multiple agents in RSMDEA can be distributed and parallel, and so they can work with each other more successfully. The distributed and parallel algorithms to direct multiple agents can be co-evolutionary algorithms [20]. Besides, each agent of RSMDEA is multi-dimension education agent: teacher role, student role, and administrator role. So computing and data flow between one role and another role, or between one agent and another agent are very complicated [21-26]. How to acquire and keep robustness of RSMDEA is a hard problem. But with ideal robustness reducing model and real robustness relativity model based on data relative methods the problem can be solved by the distributed and parallel mechanism and algorithm. So the education network based on RSMDEA can be kept robust, and then teachers, students and administrators can benefit more from RSMDEA.

2. ARCHITECTURE OF RSMDEA

In the E-Learning system, teacher, student, and administrator are the main roles of human and intelligent agents. In fact, student role becomes increasingly important in education network today. The main body of the RSMDEA is distributed multi-agent system. The distributed system communicates with all user bodies through mobile agents and controls web information on the user bodies through mobile agents too. Education web information can be provided from the education center down to the user bodies. On the other hand, user information and knowledge requests can pass from the user bodies up to the education center, then the education center or the middle bodies of the RSMDEA can carry out some strategies and control rules to satisfy the requests from the user bodies.

Multi-dimension education agent

Practices of developing E-Learning application system indicate that, teacher, student, and administrator three roles are necessary and relative sides of education information systems, and all of the three sides show the whole effect of E-Learning. Therefore, when the model of multi-dimension education agent is built, teacher, student, and administrator three roles can be taken as three dimensions of the agent, and all of the three dimensions together implement the whole function and realize the whole characteristic.

The model of multi-dimension education agent is composed of student module, teacher module, administrator module, and common module. Moreover, student module is the core of multi-dimension education agent, because students are the main body of E-Learning, and only if students request, use,

and manage the education resources actively, the creative thinking of students will be developed, so the advantages of E-Learning can be exhibited fully. The common module refers to the common equipments of multi-dimension education agent model, including interfaces, knowledge bases, optimal algorithms, sensors and so on.

Why are agent techniques necessary for distributed E-Learning system? At first, intelligence has become one of most important criterion to develop distributed E-Learning system, not only because intelligence is necessary for the development of E-Learning, but also because intelligence can improve the performances of distributed system, such as cooperation and learning. Agent technique has become one of most important and burgeoning intelligent techniques. Second, the development of independent component technique, a new powerful programming method, provides excellent base to develop and implement agents. Besides, agents are fit for using evolutionary algorithm, which is good at distributed and parallel computation. Moreover, mature agent interface has pushed customization of E-Learning network services for users. At last, immune mechanism can be introduced based on agent techniques to keep the distributed E-Learning system robust and safe.

Principles of RSMDEA

Because RSMDEA takes use of agents of three-dimension structure and co-evolutionary algorithms, RSMDEA has its own advantages and good safety in the applications of E-Learning.

1) Three-dimension structure with exchange is easy to use.

The model of multi-dimension education agent has the three-dimension structure: teacher role, student role, and administrator role. For a user, however the user is a teacher, a student, or an administrator, he can always find the appropriate interface of the model, and so the bridge between the user and the education information system can be established, and it is easy to operate.

2) It is easy to search optimally, build mobile education, distributed/parallel education, and education active network.

RSMDEA takes evolutionary algorithms as the basic optimal algorithms to realize heuristic search with strong random characteristics and develop search efficiency for information in the process of E-Learning. Openness, independence, and combinability of agents have provide excellent platform and interfaces to introduce advanced computing techniques, such as mobile agent technique, multi-agent technique, distributed network technique, parallel computing technique, active network technique and so on. So it is convenient to expand and customize user services, and then the development and individuation of E-Learning services can be developed.

3) Immune mechanisms can be introduced into RSMDEA to develop immunity and stronger safety.

Multi-dimension education agents of RSMDEA can have immunity by immune algorithm, and then RSMDEA can strengthen the ability to resist and eliminate the adventitious damages of virus, electronic bomb, bug, illegal invasion etc. So RSMDEA can develop stability, safety, and robustness of the E-Learning system.

Structure of RSMDEA

Main agents of RSMDEA and their functions are shown as the following:

1) Teaching agent

In the E-Learning system, Teaching agent is responsible for intelligent scheduling and management of various E-Learning class resources and programs. Based on agent techniques, the RSMDEA platform completely shows the education idea centering the students. And the platform create a real E-Learning environment, which satisfies the requirements of autonomous learning, and provides students with learning methods/strategies and more direct directions in the process of learning.

2) Searching agent

In E-Learning system with several large databases, searching agent is used to overcome the managing bottleneck of databases, with fuzzy matching and reasoning technique, natural language processing technique, and intelligent database management technique. The searching agent can communicate with teaching agent and other agents, and schedule together, in order to maintain intelligence, customization, and optimization of the whole distributed multi-agent system.

3) Managing agent

Many advanced education techniques such as agent, fuzzy reasoning, self-adaptation, learning navigation etc. have been applied widely, and each module of RSMDEA has some intelligent functions. According to constructionism, the advantages of Internet resource sharing and remote communication have been developed to the maximum, and the learning environment easy to construct, generate, and combine knowledge based on Internet has been advocated and established for students. With course resource bases, direct support has been made for students in contents, operations, methods, and strategies.

4) Co-learning agent

Based on the three kinds of student models, AI studying assistant has been developed, and users can select their own favorite learning pattern with the assistant and acquire instant and effective intelligent tips from the co-learning agent of assistant. Different questions in the learning process can be solved in time with synchronous/asynchronous answering systems. Students can get learning experiences and effective methods of the senior students from the AI assistant, and compare the experiences/methods with theirs to find knowledge shortcomings, which need improving and remedying. The co-learning agent can talk with students, and establish customized learning spaces for students.

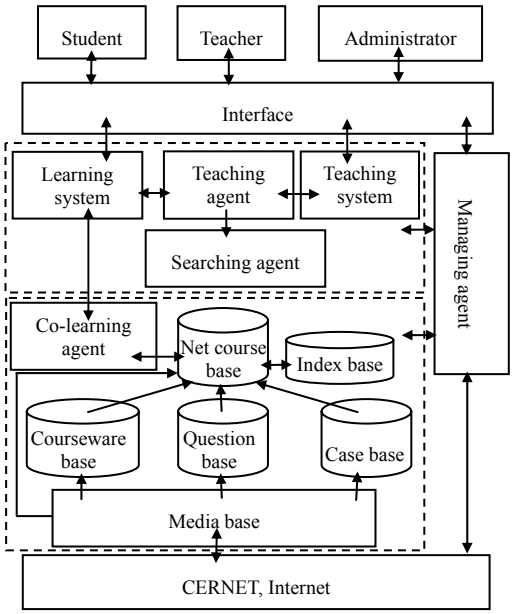


Fig. 1 Structure of RSMDEA

5) Knowledge base

Knowledge base mainly consists of knowledge point

warehouse, co-learning information base, management information base, and index base etc. Knowledge-point warehouse can be divided, by multi-media form, into text base, figure/image base, wave base, video base and so forth, and can include courseware base, question base, case base, reference base and so on, according to contents.

6) Human-machine interface
Human-machine interface can be implemented basically by XML and HTML web pages. The style of the web pages is required concise, beautiful, comforting. The language of the interface is required easy, standardized, fit for E-learning.
The structure of RSMDEA is demonstrated as figure 1.

3. PROPERTIES OF RSMDEA

Based on analyzing architecture of RSMDEA, some properties can be summarized as the following:

1) Optimal Efficiency
RSMDEA has divided functions into some independent modules. According to the principles of parallel computation, RSMDEA is fit for parallel computing. And different modules can be implemented at the same time in different computers or main machines. So time cost can be decreased in the times of division number, and RSMDEA can work extraordinarily efficiently. For example of co-evolutionary algorithms, the algorithm can be parallel-computed in RSMDEA.
Education network is a large-scale network with expanding scale and complexity, agent need to be multiplex, and the functions and computing can be distributed into each agent to build distributed/parallel computing environment. Each agent can communicate and compute interactively, and accomplish given teaching or service tasks. Cooperation between one agent and another agent can be implemented according to the evolution law, and the base of evolution is the gene of each agent. In a class of agents, the gene set of some agent cluster is called as gene pool, and agent is the loader of gene. If the environment varies, the mechanism of selection, mutation, and intercrossing will also vary. The variation will cause the variation of the gene pool. A class of gene is always experiencing isolation, gene flow, and variation recursively. Initially, a group of isolated agent clusters develop individually, and genes flow quickly and interiorly. With the opening of genes, independent education agents communicate and compete. Hence, cooperation of independent education agents is fit for parallel computing.

2) Intelligence
When the information nodes of the RSMDEA are intelligent agents, the agents can learn, communicate, and compute with intelligent evolutionary algorithms. User interface of the RSMDEA can be more interesting and friendly with agent technology. Information search can be of optimal efficiency with evolutionary algorithms. The RSMDEA will be more intelligent and interactive with multi-dimension mobile agents.

3) Immunity
When a RSMDEA is damaged by the adventitious factors, if the distributed multi-agent system has immune algorithm and its immunity becomes stronger [16-18], then it is more resistant to the adventitious damage, so its robustness becomes better; conversely, if its immunity becomes weaker, then it is less resistant to the adventitious damage, so its robustness becomes worse. Moreover, immunity of the RSMDEA is related to its learning frequency and efficiency, like the man. Hence, the better immunity the RSMDEA has, the stronger its robustness is.

4) Robustness
After immune mechanism of the RSMDEA has been formalized by information structural network, robustness of the artificial immune system can be analyzed [16-18].
Suppose a RSMDEA R is damaged from the adventitious factors, and the information entropy E_R of the system R is decreased. Then the immune algorithms of the RSMDEA R will minimize the damages and adjust inner information entropy distribution of the RSMDEA or increase the whole information entropy from outside by learning with immune algorithms. If its immunity becomes stronger, then the intelligent system is more resistant to the adventitious damage, so its robustness becomes better; conversely, if its immunity becomes weaker, then it is less resistant to the adventitious damage, so its robustness becomes worse.

5) Safety & Interactivity
Since the RSMDEA is immune against foreign attack, the system will keep safe with defense of immune algorithm.
Agents and information nodes in the RSMDEA are multi-dimension: teacher role, student role, and administrator role. One dimension of agents and information nodes can interact with another easily.

4. ROBUST MODEL OF RSMDEA

Based on problem reduction method, robust problem of discrete distributed multi-agent system in ideal distributed environment can be reduced and transformed into robust problems of single agents and discrete control processes. So problems can be simplified and research difficulty degree can be descended. Then by solving easier robust problems, robust problem of the whole distributed multi-agent system can be solved. This is to say, model and theorem on reducing robust problem of multi-agent system can be established. For real distributed multi-agent system, the model on robust relativity of multi-agent system can be created in the next section, and the relation between the reducing model and the relativity model will be analyzed.

Problem reduction method
Problem reduction method is an effective analysis method to transform complex problems to simpler problems and reduce multi-element problems into multiple single-element problems. Suppose in ideal distributed computing environment of RSMDEA, each agent accomplishes its different task individually, so robustness of each agent is independent on others. Thus, problem reduction method can be applied to robustness analysis of RSMDEA, by dividing robustness analysis to each agent and discussing robust relations between any two agents.

Model on reducing robust problem of RSMDEA
Below, based on the problem reduction method, two types of RSMDEA have been analyzed on robustness: hierarchical distributed multi-agent system and equipollent distributed multi-agent system.

Hierarchical distributed multi-agent system: In the hierarchical distributed multi-agent system, there are operation processes of high-hierarchy agent and low-hierarchy agents, and discrete control processes from high-hierarchy agent to low-hierarchy agents. The structure of the hierarchical distributed multi-agent system can be shown in figure 2.

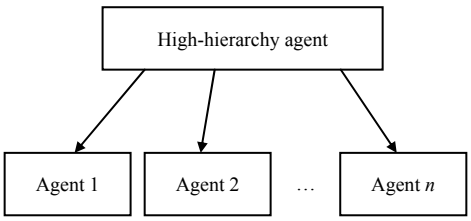


Fig. 2 Hierarchical distributed multi-agent system

With problem reduction method, robust problem of hierarchical distributed multi-agent system can be divided into three sub-problems:
1) Robustness analysis problem of high-hierarchy agent;
2) Robustness analysis problem of each low-hierarchy agent;
3) Robustness analysis problem of the discrete control process from high-hierarchy agent to multiple low-hierarchy agents.

Equipollent distributed multi-agent system: In the equipollent distributed multi-agent system, there are operation process of each agent, and discrete control process from one agent to another agent. The structure of the equipollent distributed multi-agent system can be shown in figure 3.

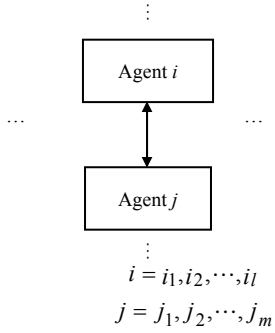


Fig. 3 Equipollent distributed multi-agent system

With problem reduction method, robust problem of equipollent distributed multi-agent system can be divided into two sub-problems:
1) Robustness analysis problem of each agent;
2) Robustness analysis problem of the discrete control process from one agent to another agent.

Model on reducing robust problem of RSMDEA: Suppose RSMDEA consists of n agents $A_i, i = 1,2,\cdots,n$ and discrete control process from one agent to another $C_{jk}, j,k = 1,2,\cdots,n$ (if RSMDEA has hierarchical structure, then RSMDEA will have a high-hierarchy agent A_0), and RSMDEA is notated as $\{A_i, C_{jk} | i = 1,2,\cdots,n; j,k = 1,2,\cdots,n\}$; robust criterion set of each agent $A_i, i = 1,2,\cdots,n$ is represented as $R_{A_i}, i = 1,2,\cdots,n$ respectively, robust criterion set of high-hierarchy agent A_0 (if possible) is represented as R_{A_0}, and robust criterion set of discrete control process $C_{jk}, j,k = 1,2,\cdots,n$ is written as $R_{C_{jk}}, j,k = 1,2,\cdots,n$, then the model on reducing robust problem of RSMDEA can be described as the following robust criterion set:

$$R_A = \begin{cases} \bigcup_{i=1}^{n} \left(R_{A_i} \bigcap_{\exists j=i \text{ or } k=i; j,k=1,2,\cdots,n} R_{C_{jk}} \right) \\ \text{if } A_0 \text{ does not exist;} \\ \bigcup_{i=1}^{n} \left(R_{A_i} \bigcap_{\exists j=i \text{ or } k=i}^{j,k=1,2\cdots,n} R_{C_{jk}} \right) \bigcap R_{A_0} \\ \text{if } A_0 \text{ exits.} \end{cases} \quad (1)$$

Theorem on reducing robust problem
Based on the model on reducing robust problem of RSMDEA and problem reduction method, it is obvious that the theorem on reducing robust problem of RSMDEA can be concluded as the following:
For ideal RSMDEA, its robust criterion can be analyzed in the following two cases:
1) When RSMDEA is hierarchically distributed, the robust criterion set of the operation process of RSMDEA {A_0, A_i, C_{0i}, here A_0 represents the operation process of high-hierarchy agent, the control process of low-hierarchy agent is written as $A_i, i = 1,2,\cdots,n$, n represents the number of low-hierarchy agents, C_{0i} represents the discrete control process from high-hierarchy agent to low-hierarchy agent.} can be

$$\{P(A_0) > 0\} \bigcap \left(\bigcup_{i=1}^{n} \{Q_i(A_i) > 0\} \right) \bigcap \{R(C_{0i}) > 0\} \quad (2)$$

Here, $\bigcup_{i=1}^{n} \{\cdot\}$ represents the union operation to the set sequence $\{Q_i(A_i) > 0\}$, robust criterion set of high-hierarchy agent can be written as $\{P(A_0) > 0\}$, robust criterion set of each low-hierarchy agent can be represented as $\{Q_i(A_i) > 0\}$, and robust criterion set of the discrete control process from high-hierarchy agent to low-hierarchy agent can be represented as $\{R_i(C_{0i}) > 0\}$;

2) When agents of RSMDEA are equipollent and distributed, the robust criterion set of the operation process of RSMDEA {A_i, C_{jk}, here Ai represents the operation process of agent Ai, C_{jk} represents the discrete control process from agent A_j to agent A_k.} can be

$$\bigcup_{i=1}^{n} \left(\{P_i(A_i) > 0\} \bigcap_{\exists k=i \text{ or } j=i}^{j,k=1,2,\cdots,n} Q_{jk}(C_{jk}) > 0 \right) \quad (3)$$

Here, $\bigcup_{i=1}^{n} \{\cdot\}$ represents the union operation to the set sequence $\{P_i(A_i) > 0\}$, robust criterion set of each agent A_i, i=1, 2, …, n can be written as $\{P_i(A_i) > 0\}$, robust criterion set of the discrete control process C_{jk} between agent A_j to agent A_k can be represented as

$$\begin{cases} \{Q_{jk}(C_{jk}) > 0\}, \text{ process } C_{jk} \text{ exists;} \\ \Phi, \text{ process } C_{jk} \text{ does not exist.} \end{cases} \quad (4)$$

Therefore, with the model and the theorem on reducing robust problem of RSMDEA, actual robust problem of real RSMDEA can be researched and solved, using the robust theories of single agent and the robust theories of discrete control process.

5. ROBUST RELATIVITY OF RSMDEA

To study robust problem of actual RSMDEA, a series of novel theory frames need to be established, as a bridge to transform the multi-element robust problem of RSMDEA into multiple single-element robust problems of agents and discrete control processes in RSMDEA.

Concepts of robust relativity:
Definition 1
Actual RSMDEA cannot satisfy all the requirements of ideal distributed multi-agent, i.e. not every agent operates individually, and some agents of RSMDEA may affect robustness of other agents to attain their own robustness. The robust criterion sets of agents are relative to some extent, and this property can be called robust relativity.
Definition 2
The coefficient to represent the relative extent between one agent and another agent can be called robust relative coefficient.
For an instance, if agent A_1 and agent A_2 are 80% relative, then the robust relative coefficient of both can be 0.8.
Definition 3
Different agents described with unified form can be called equivalent agents. And robustness of the equivalent agent is also equivalent, that is to say robustness of the agent can be described with unified form [18].

Model of robust relativity: With problem reduction method, the model and the theorem on reducing robust problem of RSMDEA in ideal distributed environment have been established, and the concepts of robust relativity have been made. Afterwards, for actual RSMDEA, the model on reducing robust problem of RSMDEA can be developed to the model on robust relativity of actual RSMDEA according to relativity between one agent and another agent, so that robust characteristics of RSMDEA can be analyzed further.
For a RSMDEA $\{A_i, C_{jk} | i, j, k = 1, \cdots, n\}$ with n element, the robust relative coefficient among n_j agents can be written as $\lambda_j, j = 1,2,\cdots,m$, and m is a constant. Robust criterion set of each agent $A_i, i = 1,2,\cdots,n$ is represented as R_{A_i}, the robust criterion set of the high-hierarchy agent can be written as R_{A_0}, robust criterion set of discrete control process from one agent to another agent $C_{jk}, j, k = 1,2,\cdots,n$ can be represented as $R_{C_{jk}}, j, k = 1,2,.\cdots,n$, and the robust criterion set of RSMDEA is written as R, then the problem of robust criterion set R for RSMDEA can be reduced into the following sub-problems:
1) The problem of robust criterion set R_{A_i} for each agent $A_i, i = 1,2,\cdots,n$.
2) The problem of robust criterion set R_{A_0} for high-hierarchy agent A_0 (if it exists).
3) The problem of robust criterion set $R_{C_{jk}}, j, k = 1,2,.\cdots,n$ for discrete control process $C_{jk}, j, k = 1,2,\cdots,n$.
By solving all the sub-problems, robust criterion sets of all the sub-problems can be calculated with the robust relative coefficient, and then the problem of robust criterion set of complex RSMDEA can be solved.
This is named as the model on robust relativity of RSMDEA, and the structure of the model is shown as figure 4. In this figure, the arrows from top to bottom represent reduction, and the ones from bottom to top represent operations and synthesis of robust criterion sets.

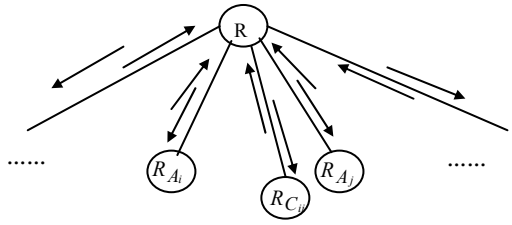


Fig.4 Model on robust relativity of RSMDEA

According to the model on robust relativity of RSMDEA, robust problems can be reduced from top to bottom, and the difficulty and element number can be decreased recursively until the problems are reduced into simpler problems or the problems which have been ever solved, such as robust problems of some single agents and robust problems of discrete control processes. Then these simpler problems will be solved; subsequently contrarily from bottom to top, all the robust criterion sets of simpler problems will be calculated with the model on robust relativity of RSMDEA, and then the robust criterion set of RSMDEA can be solved.
In nature, the model on reducing robust problem of RSMDEA is fit for ideal distributed multi-agent system, and the model on robust relativity of RSMDEA is used to analyze robustness of actual RSMDEA and relativity of robust agents in RSMDEA. Hence, the model on reducing robust problem of RSMDEA is simplified from the model on robust relativity of RSMDEA, and the latter is used to develop and expand the former. The two models come down in one continuous line: the model on reducing robust problem of RSMDEA is the base to study the model on robust relativity of RSMDEA, and the latter is the product of exploring the applications for the former. In ideal distributed computing environment, because each agent operates individually in function, the robust relative coefficient in the model of robust relativity is 0, and so the model on robust relativity of RSMDEA can be simplified to the model on reducing robust problem of RSMDEA. On the other hand, the bridge from the model on reducing robust problem of RSMDEA to the model on robust relativity of RSMDEA can be built with robustness analysis of each agent, and the theorem on reducing robust problem and the method on robustness analysis of single agent can be transformed into the tools for robustness analysis of actual RSMDEA, by creating and improving the novel frame of robust relativity theory.

6. APPLICATIONS OF RSMDEA

Prototype development AI net course in RSMDEA is a process of developing modules and expanding functions from a small system [21-22]. At first, basic contents and functions need choosing. This requires designer to understand the foundational knowledge structure of AI course based on the model of RSMDEA, so that the first step of development is to abstract knowledge-points of AI course and establish the basic knowledge structure of this course. Afterwards, the themes of this net course in every web page will be created, according to teaching and studying experiences for tens of years and characteristics of the E-Learning model, and the pace and difficulty level of AI course will be arranged. Besides, relation

between one knowledge-point and another knowledge-point, connections among the themes, and interactivity between teaching and studying are required explicit and represented with formalized models. So better E-Learning effect can be made. After formalization of the E-Learning model, functions of the E-Learning prototype can be attained with agent technique and programming languages such as Java, XML, and so forth [23-26]. This is what work will be done during the stage of programming.

Based on the model of RSMDEA and the prototype of AI online course, the following functions and properties of this prototype can be listed:

1) Online course system has been established to reflect current new frontiers in AI field.

2) Net virtual multi-media classroom and teacher office have been created based on distributed multi-agent technique.

3) Online place of extracurricular activity and communication for students has been built.

4) Effective online education management has been made and so on.

In a word, E-Learning environment can be created with the prototype of AI net course to simulate real life, teaching resources can be shared effectively on the Internet, and E-Learning can be convenient for users.

7. CONCLUSIONS

Since E-Learning was presented, AI has been the crucial technique to realize customization and individualization of E-Learning, and distributed computing technique as a way to develop high efficiency of advanced system has been combined with latest agent technique to develop advanced intelligent E-Learning platform. This platform becomes the base of RSMDEA, and the distributed multi-agent system will certainly be welcome and cared. In this paper, RSMDEA has been proposed, principles and structure of RSMDEA have been designed, and quality of RSMDEA such as robustness has been analyzed based on problem reduction method. In order to emphasize and delve robustness of actual RSMDEA, ideal RSMDEA has been analyzed, the model on reducing robust problem of RSMDEA has been established, and the theorem on reducing robust problem of RSMDEA has been presented, firstly. Afterwards, robust relativity has been defined and introduced into the robust research of actual RSMDEA, the model on robust relativity of RSMDEA has been represented, and the relation between the model on reducing robust problem and the model on robust relativity for RSMDEA has been analyzed. So RSMDEA has a robust characteristic in its applications of various fields, especially in E-Learning development and applications.

8. ACKNOWLEDGES

The supporting foundations of the research on RSMDEA include National Natural Science Foundation of China (69974043), National Doctoral Foundation of China (99053317), and National New Century Net Course Construction Engineering Foundation of China (1441000). Some advanced theory and new ideas are from results of National Natural Science research. AI online course development in National New Century Net Course Construction Engineering project provides good test and application environment for the RSMDEA. At last, thanks need to be given for good advice from anonymous referee.

9. REFERENCES

[1] Marakis J.G., Chamico J. et al, “Parallel ray tracing for radiative heat transfer: Application in a distributed computing environment”, International Journal of Numerical Methods for Heat and Fluid Flow, Vol.11, No.4, 2001, p 663-681.

[2] Garg V.K., Skawratananond C, “String realizers of posets with applications to distributed computing”, Proceedings of the Annual ACM Symposium on Principles of Distributed Computing, Aug 2001, pp.72-80.

[3] Laure E., “OpusJava: A Java framework for distributed high performance computing”, Future Generation Computer Systems, Vol.18, No.2, October 2001, pp.235-251.

[4] Kranakis E., Santoro N., “Distributed computing on oriented anonymous hypercubes with faulty components”, Distributed Computing, Vol.14, No.3, July 2001, pp.185-189.

[5] Wang Chen, Teo Yong Meng, “Supporting parallel computing on a distributed object architecture”, Journal of Systems and Software, Vol.56, No.3, Mar 2001, pp.261-278.

[6] Shaw Neal G., Mian Ahmad et al, “A comprehensive agent-based architecture for intelligent information retrieval in a distributed heterogeneous environment”, Decision Support Systems, Vol.32, No.4, March 2002, pp.401-415.

[7] Shan Mi-Yuan, Cai Zi-Xing et al, “Synergic production mechanism based on multi-agent for a multi-stage distributed manufacturing system”, Kongzhi yu Juece/Control and Decision, Vol.16, No.4, July 2001, pp. 410-414.

[8] Qi Hairong, Iyengar S. Sitharama et al, “Multiresolution data integration using mobile agents in distributed sensor networks”, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, Vol.31, No.3, August 2001, pp.383-391.

[9] Corbin Malcolm, “Applications of mobile agents and ambassadors in distributed simulation”, Simulation Practice and Theory, Vol.6, No.6, Sep 1998, pp.505-532.

[10] Shim H.-S., Kim H.-S. et al, “Designing distributed control architecture for cooperative multi-agent system and its real-time application to soccer robot”, Robotics and Autonomous Systems, Vol.21, No.2, Sep 1997, pp.149-165.

[11] Maher M.L., “A model of co-evolutionary design”, Engineering with Computers, Vol.16, No.3-4, 2000, pp.195-208.

[12] Baek Dong Hyun, Yoon Wan Chul, “Co-evolutionary genetic algorithm for multi-machine scheduling: Coping with high performance variability”, International Journal of Production Research, Vol.40, No.1, Jan 2002, pp.239-254.

[13] Tao Gong, Zixing Cai, “Building and Application of Multi-Dimension Education Agent”, China Education Info, No.76, July 2002, pp.55-56.

[14] Tao Gong, “The method of building and bettering ICAI students’ model”, Journal of Xiangtan Normal University, Vol.23, No.1, January 2001, pp.58-61

[15] Tao Gong, Zixing Cai, “Design for electronic Home-Teaching System based on ICAI model”, Computing Technology and Automation, Vol.20, No.2, March 2001, pp.47-50.

[16] Zixing Cai, Tao Gong, “Multi-dimension education immune network”, Proceedings of the 6th World Multi-Conference on SYSTEMICS, CYBERNETICS

AND INFORMATICS, July 2002, Vol.1
[17] Zixing Cai, Tao Gong, “Analysis on Robustness of Intelligent System Based on Immune Mechanisms”, Proceedings of the First China-Japan International Workshop on Internet Technology and Control Applications, June 2001, pp.56-61.
[18] Tao Gong, Zixing Cai, “Robust Analysis on Discrete Multiple Equivalent Agents System”, Computer Science, 2002, Accepted
[19] Zixing Cai, Xiang Zhou et al, “Evolutionary Control: Principles and advantages”, Proceedings of Asian Conference on Control, June 2000
[20] Zhihong Peng, Path planning of cooperative mobile multi-robot systems, Doctoral thesis of Central South University, 2000
[21] Zixing Cai, Guangyou Xu, Artificial Intelligent: Principles and Applications (Second Edition), Beijing: Tsinghua University Press, 1996.
[22] Zixing Cai, Intelligent Control: Principles, Techniques and Applications, Singapore: World Scientific, 1997.
[23] Tao Gong et al, Commerce Server Managing Expert, Beijing: Publishing House of Electronic Industry, 2002.
[24] Tao Gong et al, Internet Security and Acceleration Server Managing Expert, Beijing: Publishing House of Electronic Industry, 2002.
[25] Tao Gong et al, BizTalk Server Advanced Development, Beijing: Publishing House of Electronic Industry, 2002.
[26] Tao Gong et al, Oracle9i Database Advanced Management, Beijing: Publishing House of Electronic Industry, 2002

Towards Distributed Information Exchange: An Application from XML to OWL *

Qiwei Yin, Shanping Li, Yujie Hu, Ming Guo, Xiangjun Fu
Institute of Artificial Intelligence, College of Computer Science, Zhejiang University
Hangzhou, Zhejiang 310027, China
E-mail: qiwei-yin@163.com

ABSTRACT

Information exchange is crucial not only among computers of the same workgroup, but also among the computers distributed in all over the world on Internet.
It is difficult for computers to process large amounts of HTML-coded information resource to facilitate information exchange effectively. XML differs from HTML in that it separates data and formatting instructions. As a result, it is being adopted universally as a means of automatically exchanging data on the Internet. However, XML's lack of semantics prevents distributed computers from reliably performing information exchange on a semantic level.
Recently, ontology has become an interesting area in knowledge study. In computer science, ontology is a formal specification of a shared conceptualization. Web Ontology Language (OWL) such as DAML+OIL can be regarded as the languages based on Web standards for describing ontologies.
DAML+OIL provides a specification framework for independently creating, maintaining, and interoperating ontologies while preserving their semantics.
This article focuses on product information representation and exchange using from XML to OWL, and shows the value and significance of OWL for distributed information exchange. Lastly, a framework for OWL-based product information exchange is presented, and the peer-to-peer computing used in the framework's ontology layer is introduced.

Keywords: Semantic Web, Ontology, OWL, XML, DAML+OIL, Distributed Computing, Peer-to-Peer Computing, Product Information.

1. INTRODUCTION

Information exchange is crucial not only among computers of the same workgroup, but also among the computers distributed in all over the world on Internet.
Currently information on the internet is presented using Hypertext Markup Language (HTML). It is difficult for computers to automate the processing of large amounts of HTML-coded information resource effectively, because the HTML combines both data content and formatting instructions. While it is possible to search for patterns within sites to automatically extract data, it is time consuming and unreliable. For this reason, the HTML-based Web has not been a particularly attractive mechanism for automated information exchange among distributed computer systems.
XML (eXtensible Markup Language) [1] differs from HTML in that it separates data and formatting instructions. As a result, it is being adopted universally as a means of automatically publishing, storing and exchanging data on the Internet. Information is coded is human-readable and contains information about the actual content. Because the data is structured, it can easily be searched, aggregated, transformed or interpreted by other systems.
However, although XML (includes XML DTD and XML Schema) is sufficient for exchanging data between computers that have agreed to definitions beforehand, their lack of semantics prevent distributed computers from reliably performing this task given new XML vocabularies. The same term may be used with different meaning in different contexts, and different terms may be used for items that have the same meaning [2].
Recently, ontology has become an interesting area in AI for knowledge sharing and information exchange, etc. An ontology is a commonly agreed understanding, and in computer science, ontology is a formal specification of a shared conceptualization [3]. Web Ontology Language (OWL) [4] can be regarded as the languages describing ontologies based on Web standards.
This article focuses on product information representation and exchange using from XML to OWL, and shows the value and significance of OWL for distributed information exchange. Lastly, a framework for OWL-based product information exchange is presented, and the peer-to-peer distributed computing used in ontology layer is discussed.

2. BACKGROUNDS

In philosophy, Ontology is a systematic account of Existence. Philosophical ontology is the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality. For AI systems, what "exists" is that which can be represented. At the computer science domain, ontologies aim at capturing domain knowledge in a generic way and provide a commonly agreed understanding of a domain, which may be reused and shared across applications and groups [5]. Ontologies provide a common vocabulary of an area and define -with different levels of formality- the meaning of the terms and the relations between them. Tom Gruber's contribution in 1993 was actually that of making the first credible attempt at defining the term [3]. In his definition, "an ontology is an explicit specification of a conceptualization". In 1998, an attempt to clarify and formalize the definition further was presented by Guarino [6]. In addition, he gave a new definition "an ontology is an intentional semantic structure which encodes the implicit rules constraining the structure of a piece of reality". A formal ontology has some underlying logical structure and gives a common understanding about some field, these allow us to exchange the information on a semantic level among the distributed computers regardless their diverse and special information formats.
Many ontology languages have been developed. The Semantic web languages are evolved from traditional knowledge representation languages such as KIF-based Ontolingua [7],

* The research is supported by National Natural Science Foundation of China (grant no 60174053).

OKBC [8], OCML [9], Frame-Logic [10], and LOOM [11]. Ontology Exchange Language (XOL) [12], SHOE [13], Ontology Markup Language (OML) [14], Resource Description Framework (RDF) [15] and RDF Schema (RDFS) [16], Ontology Inference Layer (OIL) [17], and DARPA Agent Markup Language (DAML) [18] can be regarded as the web ontology languages (OWL), because they are based on Web technology and standard.

The current human-centered web is still largely encoded in HTML, which focuses mainly on how text and images would be rendered for human viewing. Over the past few years, we have seen a rapid increase in the use of XML as an alternative encoding, one that is intended primarily for machine processing. The computer which process XML documents can be the end consumers of the information or they can be used to transform the information into a form appropriate for human understanding (e.g., as HTML, graphics, synthesized speech, etc.) As a representation language, XML provides essentially a mechanism to declare and use simple data structures and thus leave much to be desired as a language in which to express complex knowledge. Recent enhancements to basic XML, such as XML Scheme, address some of the shortcomings, but still do not result in an adequate language for representing and reasoning about the kind of knowledge essential to realizing the Web ontology language vision.

RDF (Resource Description Framework) and RDFS (RDF Schema) attempt to address these deficiencies by building on top of XML. They provide representation frameworks that are roughly the equivalent to semantic networks in the case of RDF and very simple frame languages in the case of RDFS. However, RDFS is still quite limited as a knowledge representation language, lacking support for variables, general quantification, rules, etc.

DAML and OIL are two important attempts to build on XML, RDF and RDFS, and both support for variables, quantification, rules to be well suited for building the Semantic Web. In December 2000, DAML and OIL came into being DAML+OIL [19], which has been submitted to W3C.

DAML+OIL is a Web ontology language for Web resources that has been created as a joint effort of the American and European ontology communities to create a standard language for the Semantic Web. It also builds on earlier W3C standards such as RDF and RDF Schema, as showed in Figure 1.

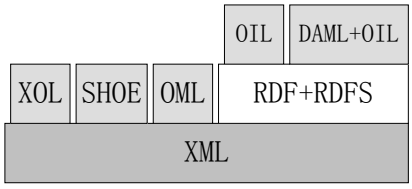


Figure 1 The pyramid of web-based languages

DAML+OIL allows representing concepts, taxonomies, binary relations, functions and instances. DAML+OIL provides modeling primitives commonly found in frame-based languages and description logic languages. As an ontology language, DAML+OIL has some features as follows:

1) It describes the structure of the domain in terms of classes and properties

2) It can assert a set of axioms such as class subsumption, class equivalence, class intersection, etc.

3) Classes can be names or expressions, and various constructors provided for building class expressions.

4) Expressive power determined by kinds of class (and property) constructor and kind of axiom supported.

W3C has recently started a new working group to develop a "web ontology language" (OWL) which should enable semantic annotations of web resources in a more formal way than provided by RDF. The OWL language uses DAML+OIL as a starting point [20].

3. INFORMATION REPRESENTATION IN XML

Recently, one of the most important techniques to facilitate the distributed information integration is XML (eXtensible Markup Language). XML is a standard method by W3C for defining structure in documents. If different data resources are formatted in XML, each computer will easily 'read' them in a uniform way. For example, we can describe a product part 'Assembly001' showed in Fig.2 using XML format as follow:

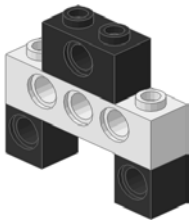


Figure 2 Assembly001

```
<Assembly ID="Assembly001">
  <Entity>TopBrick<Entity/>
  <Entity>MiddleBrick<Entity/>
  <Subassembly>Subassembly001<Subassembly>
      <Entity>RightLowBrick<Entity/>
      <Entity>LeftLowBrick<Entity/>
  </Subassembly>
</Assembly>
```

Thus, every distributed computer can read the above representation in XML. All the computers needs not to conform a special standard or a language (such as STEP and EXPRESS), if they want to facilitate the information exchange in a distributed environment.

However, it seams difficult to say that '*computer can read*' is equated with '*computer can understand*'. For example, does computer know the concept 'Assembly' encoded in XML as a tag may have the different meanings in Mechanics and Computer Science? In Mechanics, 'Assembly' can be 'a completed product part constructed by a set of manufactured parts', while it may be 'the automatic translation of symbolic code into machine code' in Computer Science. Furthermore, could computers understand the exact meanings of the word 'Assembly' defined here in Mechanics and Computer Science respectively?

It does not mean that the information resource cannot be accessible by distributed computers, but that the computers do not have the information or ability to compute with the information resource. Distributed information exchange should be upgraded from syntax (format) level to semantic (knowledge) level while conforming to Web standard.

4. INFORMATION REPRESENTATION IN OWL

Everyone has a good language education may have the ability to 'read' the materials on the Web about some field (such as CAD, Biology etc.), but 'read' is not equal to 'understand'. If a person wants to 'understand' the information distributed on Internet, he or she should have the knowledge of that field. For

example, it is a difficult thing for a person without a Mechanics or CAD background to understand the word 'Subassembly', but if he or she reads the literal definition of 'Assembly' and 'Subassembly' to have the knowledge about that, the person will thus possibly know what the 'Subassembly' is.
This situation is almost as same as to the computers. The problem is, how to let computer have the knowledge of a field? If we want to let computers understand (not only read) the information, we should give the computers knowledge. It is almost impossible to give computers knowledge written in human languages to let computers be knowledgeable. The settlement to the problem is to use the (formal) ontology.
From Guarino's definition, in the area of computer science, an ontology is an intentional semantic structure that encodes the implicit rules constraining the structure of a piece of reality. Therefore, we firstly build the ontology about the product part showed in Fig.2 to describe the knowledge of this domain explicitly.

Building the ontology
Fig.2 shows an instance example of assemblies, in fact, there are many assemblies in the CAD world. In order to make computers distributed in different locations understand and compute with all of the assemblies, we should extract the common features from them. That is to say, we should give an 'explicit specification of a conceptualization' (ontology) about the semantics of the assemblies. Thus, distributed computers can have a consistent understanding about a domain and so have the ability to exchange information among them on a semantic level.
Therefore, we firstly build the ontology named 'assembly' before describing the information resource. In the ontology 'assembly', we define two classes using DAML+OIL: 'Assembly' and 'Component', respectively referred to the whole assembly and its component:

```
<daml:Class rdf:ID="Assembly">
      <rdfs:subClassOf rdf:resource="#Product_Thing"/>
</daml:Class>

<daml:Class rdf:ID="Component">
    <rdfs:subClassOf rdf:resource="#Product_Thing"/>
</daml:Class>
```

Here, 'Product_Thing' is referred to all the things related to product. Thus, the concepts (words) we defined here, as the subclass of 'Product_Thing', can be easily distinguished from the ones in other fields.
Considering there should be two kinds of things in a component: 'Entity' and 'Subassembly', we add these two classes and define them as the subclasses of 'Component' using the keyword 'subClassOf' in the same way using DAML+OIL. However, it is not the complete definition of them. In our ontology, the semantics should be described explicitly. For example, a 'Subassembly' should have all the features of an 'Assembly', and there should be no other things in 'Component' except for 'Subassembly' and 'Entity'. Now, we define these semantics using 'subClassOf' and 'complementOf' to update the earlier definition of class 'Subassembly' and 'Entity':

```
<daml:Class rdf: about="Subassembly">
<rdfs:subClassOf rdf:resource="#Assembly"/>
</daml:Class>

<daml:Class rdf: about="Entity">
<daml:complementOf rdf:resource="# Subassembly"/>
</daml:Class>
```

Next, we define the relationship properties among the classes using 'daml:ObjectProperty'.
Firstly, the main relationship is 'Assembly has Component':

```
<daml:ObjectProperty  rdf:ID="hasComponent">
     <rdfs:domain  rdf:resource="#Assembly"/>
     <rdfs:range  rdf:resource="#Component"/>
</daml:ObjectProperty>
```

In addition, the sub relationships are:

```
<daml:ObjectProperty  rdf:ID="hasEntity">
     <rdfs:subPropertyOf  rdf:resource="# hasComponent"/>
     <rdfs:range  rdf:resource="#Entity"/>
</daml:ObjectProperty>
```

The definition of 'hasAssembly' is similar.
Last, we give a complete definition of 'Assembly' using the mechanism of 'property restriction' provided in DAML+OIL.

```
<daml:Class rdf:ID="Assembly">
   <rdfs:subClassOf rdf:resource="#Product_Thing"/>
   <rdfs:subClassOf>
     <daml:Restriction>
       <daml:onProperty rdf:resource="#hasComponent"/>
       <daml:toClass rdf:resource="#Component"/>
       <cardinality>2</cardinality>[1]
     </daml:Restriction>
   </rdfs:subClassOf>
</daml:Class>
```

Representing Information using the Concept Defined in Ontology
All the above definitions have been made in ontology 'assembly', now we can use the defined concepts to represent the information resources (Fig.2).

```
<assembly:Assembly rdf:ID="#Assembly001">
  <assembly:Entity rdf:resource="#TopBrick"/>
  <assembly:Entity rdf:resource="#MiddleBrick"/>
  <assembly:Subassembly rdf:resource="#Subassembly001">
    <assembly:Entity rdf:resource="#RightLowBrick"/>
    <assembly:Entity rdf:resource="#LeftLowBrick/>
  </assembly:Subassembly>
</assembly:Assembly>
```

'assembly:Assembly' means that class 'Assembly' is defined in ontology 'assembly' (This can avoid the conflicts among the same words in different domains that have various semantics).
The above representation style is similar to XML (XML can also fulfill the final representation), and the keywords used here only three: 'Assembly', 'Entity', 'Subassembly'; however, the most significant thing is that the implicit but intrinsic semantics of these concepts have already been defined explicitly in ontology 'assembly' in DAML+OIL. In fact, many instances of assembly can be represented using the same ontology, and distributed computers just use the 'explicit

1 Here, we restrict an assembly to have at least two components (if not, it should not be an assembly), and the component's concrete representation form is "Subassembly"or "Entity".

specification of a conceptualization' to understand the meanings of the keywords (tags or metadata), and then compute (or reason) the semantics of information resources annotated by them to facilitate distributed information exchange.

Furthermore, distributed computers can also annotate their information resources in their own format, and it does not bring any difficulties in distributed information exchange if they have the same ontology (described by a standard language such as DAML+OIL). For example, users can describe their information in XML, XOL, RDF, DAML+OIL or even database schema, and if the concepts they used are from a same ontology (or new concepts but derived from a same ontology), they will communicate each other on semantic level properly.

5. A OWL-BASED PRODUCT INFORMATION INTEGRATION FRAMEWORK IN A DISTRIBUTED ENVIREMENT

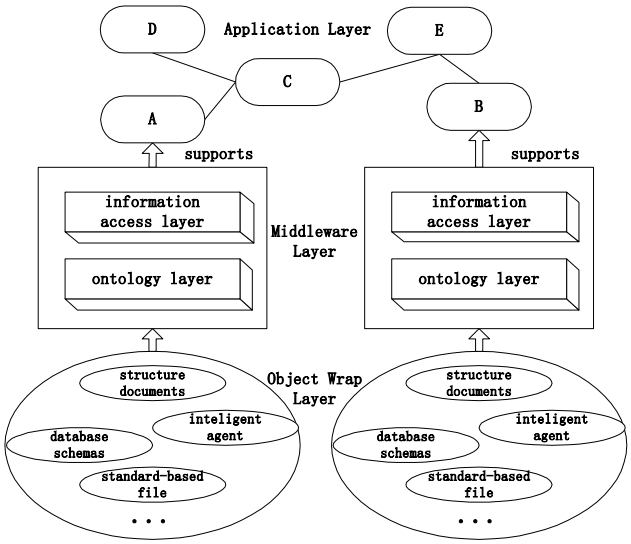


Figure 3 OWL-based product Information integration framework

As showed in Fig.3, our product data integration framework is divided into three main layers. The uppermost layer is the application layer that includes various enterprise applications, such as CAD, ERP or PDM systems. The middle layer is further divided into information access layer and ontology layer. The information access layer provides mechanism to acquire information, and we can build applications based on AGENT, CORBA or HTTP/WEBDAV technology in this layer. The ontology layer provides a way to create unified semantics of a certain product data domain using DAML+OIL, and contains a library of ontologies defined by different users. Last is the object wrap layer, in which structured or semi-structured information are annotated by concepts defined in ontologies.

Because the real value of the OWL (Web ontology language) is to let people and applications can create and discover new and interesting knowledge and share this knowledge in a transparent manner similar to the way data is exchanged today, we apply the ontology layer in a data-centered peer-to-peer distributed environment of shared and interoperable ontologies. The users will need to discover new ontologies, which are not known to them before and use them to either annotate the content or to formulate their information requests.

Peer-to-peer computing (P2P) is currently touted as the next "killer application" for the Internet. In ontology layer, P2P is used to provide a distributed architecture which can support sharing of independently created and maintained ontologies. Data Centered P2P (data-focused P2P application) allows users to search and access ontologies held on other users' systems, and synergies can be created through partnerships. In our framework, Data Centered P2P ontology layer is accomplished by creating a dynamic index server—as clients connect to the network, specific storage areas of the clients are scanned and relevant ontologies indexes are added to the server. Users seeking ontologies can search via the server, and then connect directly to other clients for direct access to the shared ontologies. When a client wants to modify or delete its own ontologies (or wants to let a shared ontology to be unshared), the index removes the reference, or availability to the reference, of the client's ontologies.

6. CONCLUSIONS

This article talks about information representation and exchange using from XML to OWL, and presents the characteristics of OWL for semantically distributed information exchange. In the end, a framework for OWL-based product information exchange is proposed, and the peer-to-peer computing used in the framework's ontology layer is introduced.

Future work will focus on building ontologies in details to cover more broad fields in product information area using OWL. Further research will pay much attention to mapping the ontologies to structured and unstructured product information resources to setup the infrastructure of product information based on ontology and the Semantic Web.

7. REFERENCES

[1] http://www.w3.org/XML/.

[2] Stefan Decker, Frank van Harmelen, Jeen Broekstra, Michael Erdmann, Dieter Fensel, Ian Horrocks, Michel Klein, Sergey Melnik, "The Semantic Web-on the respective Roles of XML and RDF", IEEE2000, http://www.ontoknowledge.org/oil/downl/IEEE00.pdf.

[3] Gruber, T. R.., "A Translation Approach to Portable Ontology Specifications", *Knowledge Acquisition*, 5, 1993, pp.199–220.

[4] http://www.w3.org/2001/sw/WebOnt/.

[5] Chandrasekaran, B.; Johnson, T. R.; Benjamins, V. R. "Ontologies: what are they? why do we need them?", IEEE Intelligent Systems and Their Applications. 14(1). Special Issue on Ontologies. Pages 20-26. 1999.

[6] Guarino, Nicola (ed.), "Formal Ontology in Information Systems", *Frontiers in Artificial Intelligence and Applications*, Amsterdam, Berlin, Oxford: IOS Press. Tokyo, Washington, DC: IOS Press, 1998.

[7] http://ontolingua.stanford.edu.

[8] V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice. "Open Knowledge Base Connectivity 2.0", Knowledge Systems Laboratory, January 1998.

[9] Domingue J., Motta E., and Corcho O, "Knowledge Modelling in WebOnto and OCML: A User Guide", http://kmi.open.ac.uk/projects/ocml/

[10] Kifer M., Lausen G., and Wu J., "Logical Foundations of Object-Oriented and Frame-Based Languages",

Journal of the ACM. 1995

[11] http://www.isi.edu/isd/LOOM/LOOM-HOME.html

[12] Karp, R. Chaudhri, V., and Thomere, J. "XOL: an XML-based Ontology Exchange Language", July, 1999. And http://www.ai.sri.com/~pkarp/xol/

[13] Jeff Heflin, James Hendler, and Sean Luke. "SHOE: A Knowledge Representation Language for Internet Applications", Technical Report CS-TR-4078 (UMIACS TR-99-71), 1999. And http://www.cs.umd.edu/projects/plus/SHOE/

[14] Robert E. Kent. "Conceptual Knowledge Markup Language: The Central Core", *the Electronic Proceedings of the Twelfth Workshop on Knowledge Acquisition*, *Modeling and Management* (KAW99). Banff, Alberta, Canada, 16-21 October 1999. And http://www.ontologos.org/OML/OML%200.3.htm

[15] Ora Lassila, Ralph R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification", *W3C Recommendation*, 05 January, 1999, http://www.w3.org/TR/PR-rdf-syntax.

[16] Dan Brickley, R.V. Guha, "RDF Vocabulary Description Language 1.0: RDF Schema", *W3C Working Draft*, 30 April, 2002. http://www.w3.org/TR/PR-rdf-schema

[17] D. Fensel et al., "OIL in a nutshell", *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, R. Dieng et al. (eds.), "Lecture Notes in Artificial Intelligence", LNAI, Springer-Verlag, October 2000. M.C.A. Klein et al. "The Relation between Ontologies and Schema-Languages: Translating OIL Specifications to XML-Schema", *Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods*, 14th European Conference on Artificial Intelligence ECAI-00, Berlin, Germany August 20-25, 2000. http://www.ontoknowledge.org/oil/.

[18] http://www.daml.org/

[19] http://www.w3.org/TR/daml+oil-reference and http://www.daml.org/2001/03/daml+oil-index.html

[20] http://www.w3.org/TR/owl-ref/ and http://www.w3.org/TR/owl-features/.

Parallel Computing Applications and Financial Modelling

Heather M. Liddell, D.Parkinson, G.S.Hodgson and P.Dzwig
Department of Computer Science,
Queen Mary, University of London,
Mile End Road, London E1 4NS, U.K
E-mail: heather@dcs.qmul.ac.uk

ABSTRACT

My colleagues and I at Queen Mary, University of London, have over twenty years of experience in Parallel Computing Applications, mostly on "massively parallel systems", such as the Distributed Array Processors (DAPs).
The applications in which we were involved included design of numerical subroutine libraries, Finite Element software, graphics tools, the physics of organic materials, medical imaging, computer vision and more recently, Financial modelling. Two of the projects related to the latter are described in this paper, namely Portfolio Optimisation and Financial Risk Assessment.

Keywords : Parallel computing, massively parallel systems, Distributed Array Processor, Financial modelling, Portfolio Optimisation, Risk Assessment.

1. INTRODUCTION

The UK has a long history of leading the world in innovative computing - this is particularly true in High Performance Computing and Communications (HPCC). Within this area there has been growing emphasis on the use of "massively parallel systems", comprising many thousands of processors working together. Two systems capable of massive parallelism are the Distributed Array Processor (DAP) and the transputer; both were invented in the UK. The DAP activity at QMW formed a nucleus for parallel processing, from which evolved a number of research groups and specialist centres in the UK ; more importantly, a large community of users was stimulated to use parallel computers for a wide variety of problems in Science, Engineering and other disciplines. High Performance Computing (and hence massively parallel processing) is of strategic national importance, both as an industry in its own right and also because of the requirements of other industries. Senator Gore, the former US Vice president stated as long ago as 1989 that "the nation which most completely assimilates high-performance computing into its economy will emerge as the dominant intellectual, economic and technological force in the next (i.e. this!) century".
The use of massively parallel systems required the development of new algorithms, software tools and application techniques; a reliable migration path must be provided from current systems - this is particularly important in the commercial world. These are the challenging areas we have been addressing in our research.

2. ALLEL COMPUTING APPLICATIONS ON THE DISTRIBUTED ARRAY PROCESSOR

The focus of our activity has always been towards users of parallel systems. We have been actively engaged in many different application areas - engineering, physical sciences, computer vision, medical imaging and econometrics. One example was an Esprit Basic Research project, OLDS [4], in which we were working with colleagues in Physics and Chemistry at QMW and with European partners, exploring the potential use of organic materials in novel micro-electronics devices. Another was a collaborative medical imaging project, MIRIAD, in which we investigated methods for the display and segmentation of features in 3D images [5], [6].
LPAC's major objective was technology transfer, primarily via collaborative projects carried out with Industrial /Commercial partners and with applications experts in our own establishments, to produce useable applications software, portable to many different parallel computing systems, and to provide migration paths to these systems. We considered the European dimension to be particularly important. We continued to address the more traditional industrial and scientific computational problem areas, including industrial modelling, use of neural networks and visualisation, but considerable emphasis was placed on the commercial and financial applications, partly because of our geographical proximity to the City of London. These were new fields for academic computing people, but we were fortunate in being able to collaborate with colleagues with expertise in these areas. The rest of this paper will concentrate on that aspect of our work, and is the result of joint research with my colleagues Peter Dzwig, Graham Hodgson and Dennis Parkinson.

3. CATION OF PARALLEL AND DISTRIBUTED COMPUTING TO PORTFOLIO OPTIMISATION

The financial markets form one of the most important engines of industry. This project is particularly interested in understanding and improving the methodology and science behind the creation and maintenance of large portfolios of financial instruments (bonds, shares etc) through the deployment of large scale numerical computing techniques.
Portfolios are collections of financial instruments whose overall intention is to, for example, minimise the risk caused by exposure to market fluctuations. From our perspective they are a collection of instruments governed by non-linear dynamics which reflect market dynamics. The requirement for the financial engineer is to select a portfolio in an optimal way. In reality such solutions are, of course, sub-optimal.
The modelling of classical portfolios and understanding their dynamics has a long history, the landmark of which was Markowitz's paper [7]. This involves creating a model in the form of a stochastic differential equation, whose stochastic terms are reflected in volatility. This volatility is conventionally taken as a constant. In prosaic terms an investor makes an estimate of what the growth of a particular component may be and this estimate becomes a parameter of the calculation. The solution to the optimisation problem can then be found by deploying conventional optimisation techniques. Naturally the growth is neither constant nor easily determined, although there may be good reason for believing that it will fall within certain bounds.

Much work has been done internationally on the modelling of individual instruments under varying conditions. In comparison little has been done on the optimisation of medium to large-scale portfolios with the use of high-density historical data sets to drive the inputs to the models. In all there may be several tens of such instruments in a portfolio (which is itself an instrument). The aim is then to model the behaviour of this problem that has perhaps fifty dimensions. In our research we set out to apply the techniques of high performance computing, in which the investigators have a great deal of experience, to the problem of modelling such portfolios.

Much has also been written in recent years concerning large systems with deterministic constraints. This has proved to be a fruitful field for the application of large scale, and in particular, parallel computing. Conventional optimisation with non-stochastic inputs can normally be completed within reasonable run-times on PCs or similar, giving flexibility when building individual models to respond to client requests. With that approach typical PC systems permit handling of moderate sized portfolios in the region of, say, two hundred instruments. The introduction of the stochastic inputs and the associated Monte Carlo simulations makes the problem one in which issues such as parallel computation come to play a conspicuous role. Hence the problem demands the application of high performance computing [8].

In our research we investigated the numerical determination of portfolios with non-stochastic constraints combined with non-deterministic inputs, and also considered the stability of the resulting portfolios in a model which has been developed by the author and her co-workers. This required the application of parallel computing. [9], [10].

Normally it is assumed that the portfolio elements will behave as "Wiener processes" and have a predicted performance which is completely deterministic; that is, a particular share or bond will have a return of x%. In our model we allowed not only for multivariate distributions but also for returns which have a range of values, whose size and characteristics are determined purely by empirical historical data.

3.1 Outline of the Problem

The relative behaviour of securities is defined by their correlation information. Traditionally a single set of inputs for growth expectations of individual portfolio components is used in a single optimisation solution. We generalised this to allow stochastic inputs to represent the distribution of possible outcomes for the returns generated by individual portfolio components. These 'scenarios' are described in terms of specified rates of return and standard deviation and the statistical behaviour of these inputs can be selected from a set of standard statistical distributions.

The results of the many ensuing Monte-Carlo simulations contain much useful information that can be extracted by Principal Component Analysis. This information can be fed back into the model to improve the results. In this paper, some discussion on the stability of the results is included.

Rather than input a constant value for the expected rate of return of each input scenario, we allow a range of values with a generalised Wiener process being used to simulate each input scenario. Generating these Wiener processes forms the first stage of a four stage process for each market, and may be summarised as follows:

- Stage1 For each simulation generate a Wiener process for each market scenario.
- Stage 2 Solve the resulting optimisation problem at points on the "efficient frontier" [9] for each simulation portfolio) and apply Principal Component Analysis to the simulation portfolios.
- Stage 3 For those points on the efficient frontier calculate an averaged portfolio over all the simulations (we call this the 'mean' portfolio).
- Stage 4 Calculate mean and volatility of rates of return for simulated scenarios and solve again to obtain an optimum market portfolio (we call this the 'efficient' portfolio). n this way a mean portfolio and an efficient portfolio is obtained for each market. We permit instruments to be grouped into a number of separate markets, for which historical correlation information between the securities of each such market is available. An overall solution can then be obtained
- Stage 5 A further optimisation stage (across markets), optimised by constraining the proportions of each market in the overall portfolio and by using correlation information generated by the Monte-Carlo simulations across the markets.

Securities are considered as forming a single market if correlation information is available across that set of securities (see 2.2). For each market, scenarios defining the anticipated performance of the market are specified by the user; each scenario directly influences the performance of one (and only one) sector of the market, although historical correlation information is required for the whole market (so sectors need not be independent).

During our research we investigated the numerical determination of portfolios with non-stochastic constraints combined with non-deterministic inputs, and also the stability of the resulting portfolios.

3.2 The Application of HPC

We have seen that the basic inputs to the model are securities which are grouped into markets – for each market there are a number of scenarios which define the anticipated performance of that particular market. Each market is conceived of as a number of sectors; each sector may be influenced by different constraints. Hence the user can exert fine control over how each market is expected to perform. Individual markets are processed separately. An overall solution is obtained by applying the optimisation to the markets themselves.

Optimal portfolios calculated for each market are influenced by sets of constraints, which may be simple bounds or linear or non-linear constraints. The various types of constraint permit different optimum portfolios to be chosen on the efficient frontier; for example, a portfolio with minimum risk, or one with maximum return, or one with some intermediate choice of risk or return. Because we are performing many Monte Carlo simulations, we generate sets of optimal portfolios and sets of efficient frontiers. This technique has allowed us to predict portfolio behaviour that performs extremely well under most cases.

We have deployed a variant of PCA analysis to enable us to identify the behaviour of the portfolios under all the scenarios generated. In particular we have been able to use the technique to examine the components of the portfolios and their variation as the market behaviour changes. This has enabled us to identify various categories of portfolio component and to observe the role that they play in determining the behaviour of the portfolio as a whole.

The models under investigation can include differing types of securities including shares, bonds and options in (potentially geographically) diverse markets. These are defined by their correlation information. Traditionally a single set of inputs for

growth expectations of individual portfolio components is used in a single optimisation solution. We generalise this to allow stochastic inputs to represent the distribution of possible outcomes for the returns generated by individual portfolio components.

3.3 Monte Carlo Simulations

A choice of statistical distribution for the simulated scenarios is provided. Each scenario in a simulation run is considered as a generalised Wiener process; the number of discrete time steps of each scenario is specified by the user. The stochastic values calculated for each index scenario are compared with the historic values calculated from the expected rates of return and standard deviations supplied for individual securities (in the relevant sector if related scenarios are provided for a market). The ratios so obtained are used to scale the historic rates of return and standard deviations of individual securities and optimum portfolios are then determined using the scaled values.

For example, consider a market where the given scenarios are the securities themselves. The (historic) correlation matrix of the market securities is also the correlation matrix for the scenarios. The covariance matrix is formed from the correlation matrix by scaling each row and column by the standard deviation of each security. We calculate a set of scaling factors for the first simulation as follows:-

(i) each Wiener process for a scenario/security price S over time t is of the form

$$\frac{\Delta S}{S} = \mu \Delta t + \sigma \varepsilon \sqrt{\Delta t}$$

where μ is the expected growth rate of the security per unit time, σ is the standard deviation of the security price and ε is a random number from a multivariate normal distribution (since the scenarios are related).

(ii) Having generated all the Wiener processes for the simulation, we can calculate the perturbed values (growth rate and standard deviation) which result for each scenario.

(iii) We then scale the historic correlation matrix by these standard deviations to produce the perturbed covariance matrix for the simulation and solve the appropriate optimisation problems.

(iv) This process is then repeated for each simulation.

A general-purpose optimisation routine is used to determine an optimum portfolio, based upon the SQP (Sequential Quadratic Programming) method [11]. For each market, constraints may be supplied in three forms: simple bounds, linear constraints of the form $Ax \geqslant B_l$ and $Ax \leqslant B_u$, and smooth non-linear constraints (defined by a vector of constraints or optionally its Jacobian).

Constraints will often need to be included to produce a balanced portfolio across the market. Without some constraints on the proportion of certain securities, the portfolio can become dominated by a very few securities (especially for the case of maximum return). Constraints of this form can result in discontinuities of the efficient frontier. It is also usual to constrain the sum of the proportions of each security in the portfolio to one, i.e. $\sum_j X_j = 1$.

For each simulation run, five types of optimisations can be solved: the optimisations for minimum risk and for maximum return (which represent the end points of the efficient frontier); optimisations of intermediate risk and intermediate return; or other intermediate points on the efficient frontier defined using the parameterised objective function $\sum_i \sum_j X_i X_j C_{ij} - \lambda\left(\sum_j X_j E_j\right)$. This Monte Carlo simulation is repeated a specified number of times. Sufficient simulation runs should be performed to ensure the stochastic results have converged. Variance reduction techniques are used to improve convergence. A further set of optimisations are then solved using the observed means and standard deviations; analogous results are also calculated using the user-supplied values. Principal Component Analysis is then applied to the recorded portfolios. More details are given in [10]

3.4 Parallelisation of the system

There are two ways in which we can take advantage of the parallel aspects of this system.

First, the numerical algorithms used have substantial vector and matrix operations inherent in the linear algebra of the calculations. By utilising computationally efficient BLAS (Basic Linear Algebra Subroutines) in the numerical algorithms, we can ensure advantage is taken of the architecture of the hardware platform used (for example, matrix operations use optimum cache sizes and vector operations use pipelining efficiently). The effect of using such machine specific BLAS on the system is an approximate three-fold increase in computational speed on Intel PCs.

Second, the Monte-Carlo simulations are independent of one another and therefore can be executed in parallel without synchronisation. Currently available library software is not thread-safe for some of the numerical algorithms we require. A solution is to implement the system using PVM (Parallel Virtual Machine, see Geist et al [12]), which divides the computation into separate system processes that communicate by message passing. Separate library routine calls cannot then interfere with one another since they each have their own address space. The danger of the PVM approach is that the communication costs can become significant if message passing is too frequent.

Our application is ideally suited to the PVM approach. A master process controls the generation of the Wiener processes and the perturbed scenario inputs, it then spawns the appropriate number of slave processes. Each slave process is responsible for performing a number of simulations. The master process passes the perturbed scenario inputs and other historic data necessary for the optimisations to each slave process (only one message per slave process is needed for this). It then waits for the slaves to complete their computations and pass back the simulation portfolios (again only one message per slave is needed for this). The master process can then continue and analyse the results. Thus frequent message passing is avoided and synchronisation is kept to a minimum.

Whilst originally we believed that parallel computing would provide the only way forward for these calculations, recent development in PC-level architectures and performance level has meant that large-scale high power PCs can be coupled to offer an effective solution for some classes of problems.

Average computation times for each time-point of the set of tests described below (2000 simulations over 59 securities for one ‘point’ on the efficient frontier, but excluding the PCA analysis) were reduced from 13.2 minutes for the single process solution to 9.1 minutes on a twin processor Intel PC. This is a significant saving bearing in mind that the Wiener process initialisation time is not dependent on the number of points on the efficient frontier, hence multiple solutions on the efficient frontier will show more favourable savings in computational costs for solution.

3.5 Principal Component Analysis

We want to develop tools for understanding the composition and stability of portfolios under various assumptions about

how errors in risk and return are measured.
Principal components are defined as linear combinations of the individual variables appearing in the portfolio. Thus for the securities X_k, the first principal component is of the form $Y_1 = \sum_k a_{1k} X_k$ with a_{1k} chosen to maximise the sample variance of Y_1 over all of the simulation portfolios subject to $a'_1 a_1 = 1$. Further principal components Y_j are defined similarly, but are also required to be orthogonal to each of the previous principal components, thus $a'_j a_i = 0$ for i < j , [13].
Principal component analysis is essentially used to reduce the dimensionality of a given problem. The principal components are chosen so that the first component has the greatest impact on the analysis and each subsequent component has a decreasing impact. We are aiming to project the original data space onto a lower dimensional space, whilst not losing too much information. To do this we need to be able to measure the errors we are introducing by selecting the first r (≤ n) principal components.
Applying principal component analysis to the covariance matrix (of the portfolio simulations) we see that the total variance of the original system is the sum of the diagonal elements $\omega_1^2 + \omega_2^2 + ... + \omega_n^2$ or the trace of the matrix, which we denote as S_n^2. The variance of the system that is 'explained' by the first r principal components is given by S_r^2. A traditional measure of the 'goodness-of-fit' of the subspace projection defined by the first r principal components is the proportion S_r^2 / S_n^2. This measure turns out not to be well-suited to our application, hence we have designed three alternative measures based on Euclidean distance, the error in risk, and the error in return. These measures are defined either as means (ERR_1, ERR_2 and ERR_3) or in least-squares form (ERR_4, ERR_5 and ERR_6) over the simulations [10].

4. ANCIAL RISK ASSESSMENT FOR PORTFOLIO MANAGEMENT

This project aimed to build an understanding of the stability of portfolios using non-linear optimisation methods with stochastic inputs to develop the understanding of appropriate stability criteria in conjunction with a large international bank as "end users". Real market data (March 1997 – March 2001) was used to address the stability of solutions of these problems to variations in the holdings. This is important where the holder wishes to understand the sensitivity of the portfolio holdings to minor variations in the holdings. These may correspond to proportions of holdings or to variations in possible scenarios. It will also develop further the understanding of real issues in the solution of this class of non-linear optimisation problems on HPC systems in the context of real economic systems.
Portfolio managers would like to choose portfolios containing a mix of securities so that risk is controlled. The future values of securities are uncertain and these values will change relative to each other depending on future events. A UK portfolio manager might want to choose a portfolio which will follow the future market independent of the choice of when, if ever, the UK joins the EURO. There are thus a number of scenarios which a manager might postulate, each giving a different value and variance of that value independent of the actual scenario for any given security. The task of the manager is to suggest a portfolio which independent of the actual scenario, will meet some given criterion, e.g. minimum risk, maximum return or some intermediate state.
The above process was tested with equity data based on the FTSE 100 Index on the London Stock Exchange. Even this relatively small data set has brought to light a number of detailed issues relating to the sensitivity which have to be investigated if the above process is to be applied more widely and to much larger markets. The use of Principal Component Analysis can highlight which predictions are important (and which are largely irrelevant) to the portfolio optimisation. We have thus created a numerical approach for the inclusion of predictions of expectations of return in portfolios, along with linear and non-linear constraints.

4.1 Alternative Definitions of Risk

The traditional definition of risk is $\sum_i \sum_j X_i X_j C_{ij}$, where X_j is the proportion of security j in the portfolio, and $C_{ij} = \rho_{ij} \sigma_i \sigma_j$ is the covariance between securities i and j . A portfolio manager may well compare performance with some benchmark portfolio (with components B_j), hence it is natural to modify the above definition, so risk is defined as: $\sum_i \sum_j C_{ij} (X_i - B_i)(X_j - B_j)$. Another use for this type of definition is where the benchmark represents an existing portfolio, hence the use of this form of definition gives a bias towards the existing portfolio.
The need to determine various measures of risk and return arises because of the non-deterministic inputs that our approach allows. The tool that we use to achieve this is again Principal Component Analysis .

5. THE ANALYSIS OF BOND PORTFOLIOS

In order to verify our approach we constructed a series of portfolios. The remaining sections of this paper briefly discuss the results obtained.

5.1 The Portfolio data

Tests are based upon bond index data collected over a period of approximately five years. The data is close of day prices for bond indexes with five different sets of maturities (from short to long dated) across internationally diverse markets (e.g. USA, UK, Japan, Australasia and Europe), together with deposit rates and currency exchange rates. The period covered by the data runs from 1993 to March 1998. The data was initially pre-processed to remove spurious records from the time series and to interpolate any missing values. Apart from this little else was done in the way of preparation of the data and we have no reason to believe that it is other than a representative set of data.
Figure 1 shows the absolute performance of the selected portfolios of bond instruments, but with risk defined relative to a typical benchmark (with maturities weighted towards medium dated bonds).
The types of optimisations solved are maximum return and three intermediate risks (1/16, 1/8, 1/4 of maximum risk). Also an upper limit of 5% is imposed on the proportion of an individual instrument in the selected portfolio. The quarters used are 1 December 1996 to 1 December 1997, with predictions based on historical observation of each preceding quarter or on actual growth over the quarter. Both sets of

results show positive performances, but the value of good predictions is clear.

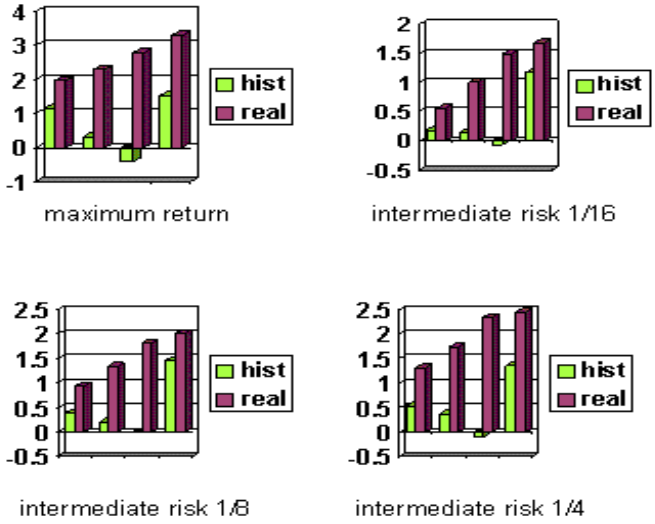


Figure 1 Bond performance with risk relative to benchmark

5.2 Sensitivity of Portfolio to Time-Period of the Historic Data

The periods, from which data is taken, that are used in this process are very long. This has necessitated Portfolio. The longer the time series used, the less relevant to more recent portfolios are the data points obtained from the beginning of the series. In order to deal with this problem we have looked at various methods of assigning significance to data points. The one we have chosen is the scheme due to Joubert and Rogers [14], in which a geometric weighting is applied.

It is clear that the choice of weighting scheme must not be such as to substantially reduce the significance of recent major events or to completely obviate those which may have occurred during the period covered. We have tested this and other models and are satisfied that the model chosen and the detailed weighting scheme applied reflects this need..

Constraints have also been applied to our modelling system. A less risky portfolio can be achieved through diversity. We have therefore chosen to limit bond holdings to a minimum of 0% and a maximum of 20% of the portfolio. Deposit holdings are bounded by ±1 (so "hedging" is allowed), but additional (linear) constraints require that (positive) deposit holdings must be matched by bond holdings and the sum of the deposit holdings is zero to limit currency speculation and to prevent gearing.

5.3 Principal Component Analysis with Predictions from Historic Data

The dataset used for the principal component analysis is that specified in above. There are 59 instruments selected, comprising 10 countries (Australia, Germany, Canada, Denmark, Japan, New Zealand, Sweden, Switzerland, UK, USA) with 10 deposit instruments and 49 bond index instruments. Points in time have been chosen at monthly intervals from March 97 to March 98. For each time-point geometric weighting (see 4.2) has been applied to the dataset to obtain the historic returns, standard deviations and covariance matrix for the 59 instruments. Significant variation in behaviour of the instruments can be observed over this period.

The stochastic inputs for these tests have been set to the historic returns and standard deviations (with geometric weighting) calculated at each of the time-points. Figure 2 shows two typical sets of simulation portfolios. Essentially there are three types of instruments present in these portfolios:- the 'popular' securities which are present in most portfolios; the 'unpopular' securities which are usually absent; and other 'volatile' securities which are present or absent because of the volatility built into the model by the standard deviations of the stochastic inputs. The first plot shows a limited number of popular and unpopular securities, with a large proportion of mid-height volatile securities; the second plot shows the portfolios concentrated into the more popular securities. We will see these two plots represent the most volatile and least volatile respectively of the time-points we have considered; this will be confirmed by the principal component analysis.

Each simulation produces an efficient portfolio determined by the perturbed input values. These efficient portfolios have a limited number of instruments in each portfolio, but not necessarily the same volatile securities present in each such portfolio. On the other hand, the mean portfolio constructed by taking means of each instrument over all the simulations, has many more instruments present – the popular securities as well as all of the volatile securities. The definition of the mean portfolio ensures it is feasible (with respect to the bounds and linear constraints), but it is not necessarily efficient.

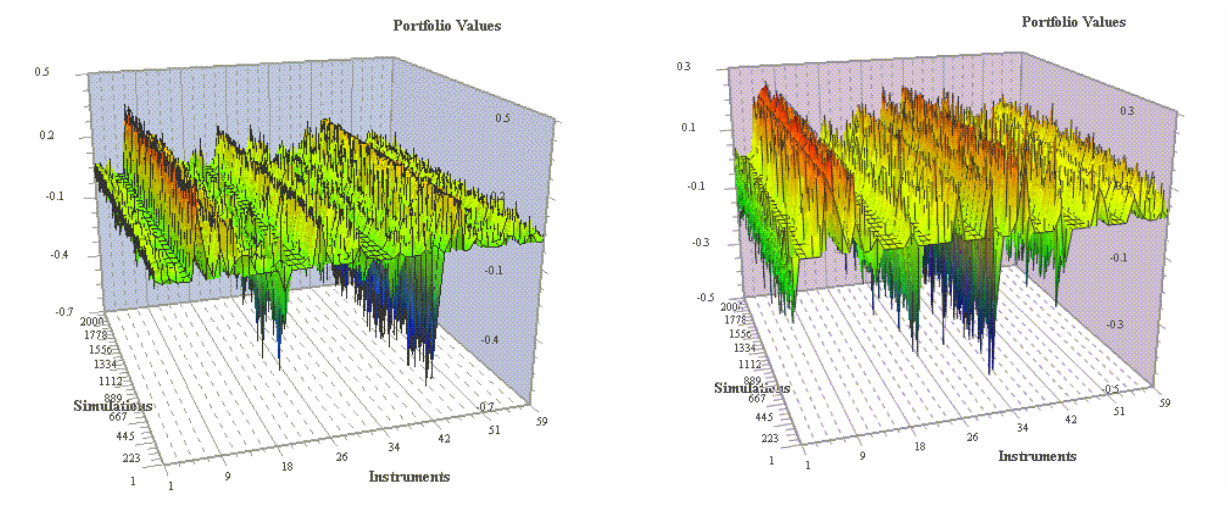


Figure 2 Typical sets of simulation portfolios

Figure 3 shows three typical simulation portfolios; different portfolios clearly show different securities present and volatility in the proportions present. The second plot shows the corresponding mean portfolio, which has many more securities present than the corresponding efficient portfolio (the conventional Markowitz optimal portfolio generated by the means of the scenario inputs).

5.4 Refinement of the Mean Portfolio

Having performed the simulations, the mean rates of return and the mean of the standard deviations observed for the simulated scenarios (the Wiener processes) are calculated. A further optimisation problem is then solved using the observed means. The resulting portfolio is of course 'efficient'. We also calculate the mean portfolio from the simulation portfolios; the mean portfolio is not, in general, efficient. The efficient portfolio will be mainly composed of the 'popular' securities, while the mean portfolio will also contain many of the 'volatile' securities.

Principal Component Analysis will have suggested that a number of the original instruments can be safely ignored. The 'unpopular' securities will have zero or very small holdings in the mean portfolio. Clearly, a first step in removing the unwanted securities from the mean portfolio is to remove the unpopular securities with values below some small threshold (uneconomic holdings).

The effect of removing such holdings and adding constraints as above appears to be to bring the iterated mean portfolio closer to the iterated efficient portfolio in risk/return space. Table 1 shows a typical example of this behaviour by repeating the simulation runs so that three sets of portfolios are available. Subtle variations in the efficient portfolios occur; analogous subtle variations can also be observed in the impact in risk/return space, where the risks and returns of the efficient and mean portfolios can be seen to converge rapidly.

Table 1

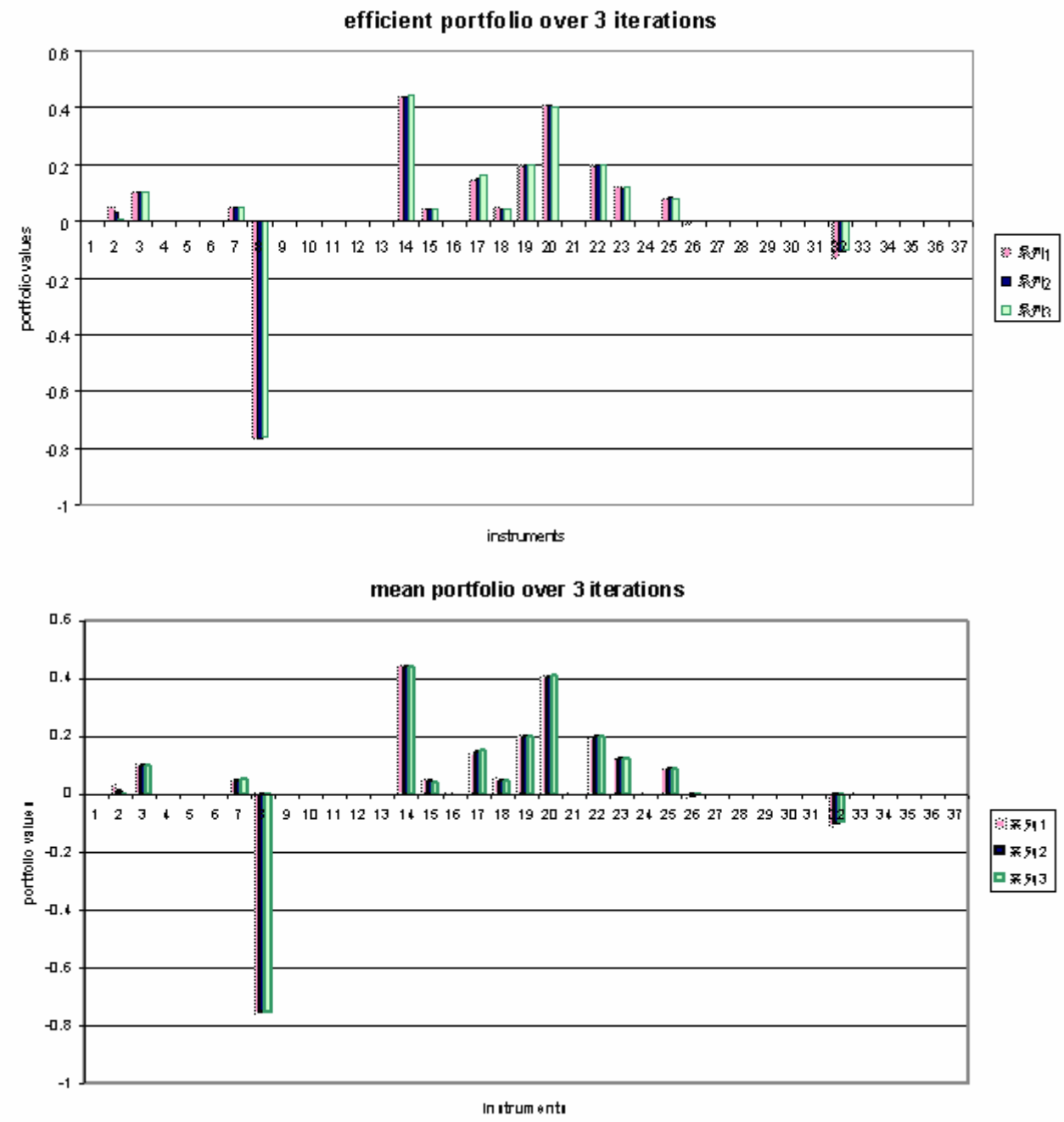
Risk of efficient portfolio	7.90E-04	7.43E-04	6.87E-04
Risk of mean portfolio	7.04E-04	6.97E-04	6.78E-04
Return of efficient portfolio	0.43641	0.42894	0.42299
Return of mean portfolio	0.42797	0.42388	0.42108

6. CONCLUSIONS

We have created a numerical approach for the inclusion of predictions of expectation of return in portfolios, along with linear and non-linear constraints. We believe that we have been able to demonstrate that the method as described is capable of producing reliable portfolios, which are stable against perturbation indicating that the selection is at least close to optimal. The timing of our tests indicates that in order to obtain sufficient convergence it may be necessary networked high-end PC technology, or other more powerful parallel architectures, for the solution of large problems. The technology has demonstrated its capability to include scenarios as inputs, dependent upon expected outcomes, and thus enhances the armoury of the portfolio manager.

7. ACKNOWLEDGEMENTS

We would like to acknowledge the invaluable input from Paul Conyers and Dominic Kini of ABN Amro and from Bernard Fromson formerly of ABN Amro. We also thank our former colleagues at the London Parallel Applications Centre, Queen Mary College and at Kings College. We would also like to thank NAg Ltd. for their assistance, in particular David Sayers. This work has been supported by the UK Engineering and Physical Sciences Research Council under grant number GR/M84282.



8. REFERENCES

[1] H. M. Liddell and D. Parkinson, "The measurement of performance on a highly parallel processor". Trans. on Computers C-32, 32 (1982)

[2] H.M.Liddell and G.S,J.Bowgen, "The DAP Subroutine Library", Comput Physics Communicatuions, 26, 311 (1982)

[3] H.M.Liddell and C.H.Lai, 'Finite elements using long vectors of the DAP', Parallel Computing 8 (1988) 351-361

[4] Organic Low Dimensional Structures (OLDS) ESPRIT BR 3200 30 months from 1/7/8 (collaborative with Universities of Edinburgh, Genoa, Max-Planck-Institute, Tubingen and Ernst-Leitz-Wetzlar GmbH)

[5] "Multidimensional and Multimodal medical imaging and display" (1989 – 91) collaborative IED project SERC GR/F34909

[6] H.M.Liddell, D. Parkinson, P.Undrill et al., "Integrated presentation of 3D imagery and anatomical atlases, using a parallel processing system" , SPIE Vol. 1653 Image Capture, Formatting and Display (1992).

[7] Markowitz H.M. 1952, "Portfolio Selection", Journal of Finance, 7, 1, p. 778.

[8] S.A. Zenios, (Editor) "Financial Optimization", Cambridgee University Press 1993

[9] G.S. Hodgson , P.Dzwig, Liddell H.M. and D. Parkinson 1998, "Application of HPC to Medium-size Stochastic Systems with Non-linear Constraints in Finance", Lecture Notes in Computer

Science, Vol. 1401, pp 411-418
[10] G.S.Hodgson,, H.M. Liddell , P. Dzwig and D. Parkinson, 2000, "Medium-size Stochastic Systems and Principal Component Analysis in Portfolio Optimisation", paper presented to the European Working Group on Financial Modelling Meeting, New Y ork, November 2000.
[11] E. Geist et al ,1994 "A users guide and tutorial for networked parallel computing"
[12] P.Gill, W.Murray and M. Wright, 1981, "Practical Optimisation", Academic Press
[13] W.J. Krzanowski, 1990, "Principles of Multivariate Analysis", Oxford University Press
[14] A.W. Joubert and L.C.G. Rogers, 1994, "Ticking over in real time" in "Quantitative andComputational Finance", p199, UNICOM 1994

Agent-based Grid Computing*

Zhongzhi Shi, Mingkai Dong, Haijun Zhang, Qiujian Sheng
Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences
PO Box 2704-28, Beijing, 100080,China
E-mail: shizz@ics.ict.ac.cn

ABSTRACT

Establishing grids is an important undertaking in developing scalable distributed infrastructures. In this paper we have proposed a model for agent-based grid computing from the implement point of view. Based on the model agent-based grid computing system AGEGC has constructed by MAGE which is a multi-agent environment platform. AGEGC has applied to develop several application systems presented in the paper. We believe that AGEGC will be useful platform for research on semantic grid.

Keywords: Grid Computing, Multi-agent Environment, Agent-based Grid Computing System

1. INTRODUCTION

Research in "Grids" has become an area of active interest. The "Grid" is an emerging infrastructure that connects multiple regional and national grids to create a universal source of computing power—the work "Grid" was chosen by analog with the electric power grid, which provides pervasive access to power [Foster 1999] The origin of the term is believed to have been the CASA project which worked on linking supercomputing sites known as meta-computing [Catlett 92].The first generation grid systems involved solutions for sharing high performance computing resources. The Information Wide Area Year(I-WAY) project is a representative first generation grid system which the virtual environments, datasets, and computers used resided at seventeen different U.S. sites and were connected by ten networks. The I-WAY project was successfully demonstrated at SC 95 in San Diego and defined following applications: supercomputing, access to remote resources, virtual reality, and video, Web, GII-Windows.

Web Services provide a means of interoperability that was essential to achieve large-scale computation with a focus on middleware to support large scale information processing. In a Grid, the middleware is used to hide the heterogeneous nature and provide users and applications with a homogeneous and seamless environment by providing a set of standardized interfaces to a variety of services. We can regard this type of grid systems as the second generation, such as Globus [Foster 97], Legion [Grimshaw 97].

In order to build new grid applications the emphasis shifts to distributed global collaboration, metadata and service oriented approach which are three key characteristics of the third generation grid systems. There is a strong sense of automation in the third generation grid systems as follows:

- Containing detailed knowledge of its components and status;
- Constructing system dynamically;
- Seeking to optimize its behavior to achieve its goals;
- Being aware of its environment.

Agent-based computing is particularly well suited to a dynamically changing environment which is an important property for the third generation grid [Roure 2002]. The agent-based computing paradigm has following features:

- Autonomy - agent operate without intervention;
- Social ability – agents interact each other using an agent communication language;
- Goal driven – agent exhibit goal-directed behavior;
- Reactivity – agents perceive and respond to their environment.

Hence we can view the Grid as a number of interacting agents. In terms of the idea we have built a prototype of agent-based grid computing (AGEGC). The next section will discuss the four layer of model for Grid. Multi-agent Environment MAGE is introduced in Section 3. The key issues of AGEGC is discussed in Section 4. There are two applications, such as e-business, oil supply chain, which will be introduced in Section 5. Finally we will point out the future work.

2. A FOUR LAYER OF MODEL

From conceptual point of view a three-layer model for the Grid infrastructure was proposed by Jeffery in a strategy document in 1999 [Jeffery 99]. In this model the computing infrastructure consists of three conceptual layers: data/computation, information and knowledge. The data/computation layer is a lower layer, which primarily concerned with computational and data resources. It is characterized as being able to deal with large-scale data, providing fast network and diverse resources as a single meta-computer. This layer builds on the physical grid fabric concerned with a distributed collection of files, databases, computers and devices.

The information layer means the information is represented, stored, accessed, shared and maintained. Here information is understood as data equipped with meaning. Uniform access to information sources relies on metadata to describe information and integrate heterogeneous sources. There are following functions in the information layer:

- Connecting together the major information sources
- Interfaces: homogeneous access to heterogeneous distributed information
- Sophisticated statistical analysis / reduction techniques for floating point numbers, textual information and multimedia information
- Special facilities for image processing, visualisation and virtual reality

The knowledge layer is concerned with the way that knowledge is acquired, used, retrieved and maintained. Here knowledge is understood as information applied to solve a problem and achieve a goal or decision-making. We can see

* This project is supported by High-Tech Programme 863 (2001AA113121) and National Natural Science Foundation of China (90104021).

that following functions will be contained in the knowledge layer:

- Acquire knowledge through KDD (knowledge discovery in database) technology of which a well-known component is data mining;
- Apply knowledge to support intelligent assists to decision makers;
- Provide interpretational semantics on the information.

Here, we have proposed a four-layer model for agent-based grid computing from the implement point of view:

(1) Common resources: consist of various resources distributed in Internet, such as mainframe, workstation, personal computer, computer cluster, storage equipment, databases or datasets, or others, which run on Unix, NT and other operating systems.

(2) Agent environment: it is the kernel of Grid computing which is responsible to resources location and allocation, authentication, unified information access, communication, task assignment, agent library and others.

(3) Developing toolkit: provide developing environment, containing agent creation, distributed computing, collaborative applications, problem solving, negotiation support, to let users effectively use grid resources.

(4) Application service: organize certain agents automatically for specific purpose application, such as power supply, oil supply e-business, distance education, e-government.

3. MULTIAGENT ENVIRONMENT

3.1 Introduction of MAGE

MAGE (Multi-AGent Environment) is an agent-oriented environment for software designing, integrating and running. It provides a new computing and problem-solving paradigm in terms of agent technology. MAGE has facilities to support agent mental state representation, reasoning, negotiation, planning, cooperation and communication. It provides system designing support, description and assembling of knowledge and capability, negotiation and cooperation designing for agent-based computing on the Internet. We spent more than ten years developing MAGE system, and now MAGE has reached Version 2.0.

MAGE is a software framework fully implemented in Java language. First it simplifies the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA specifications and through a set of tools that supports the debugging and deployment phase; Second it simplifies integration of applications through multiple schemes of software reuses and an agent-oriented software design with a graphic user interface, and also it simplifies the running management through a powerful GUI with many run-time supportThe agent platform can be distributed across machines (which not even need to share the same OS) and the configuration can be controlled via a remote GUI. The configuration can be even changed at run-time by moving agents from one machine to another one when required. The system needs only Java Run Time version 1.2 or later version as a running environment.

MAGE is being used by a number of companies and academic groups, such as Legend , Academy of Electric Power and many others. Further details and documentation can be found at http://www.intsci.ac.cn/mage.

3.2 Work flow of MAGE

According to software engineering requirements we have developed a whole procedure for software developing and system integrating in terms of MAGE, which includes requirement analyzing, system designing, agent generating and system implementing. Fig 1 demonstrates the work flow of software system developed by MAGE.

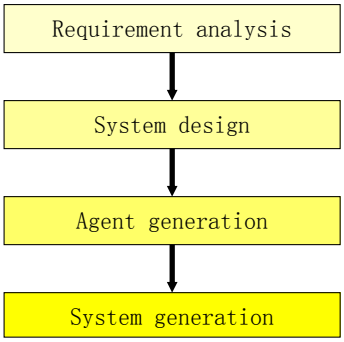


Fig 1 Workflow of MAGE

3.3 MAGE platform architecture

Fig 2 demonstrates the architecture of MAGE platform, which consists of the following components, each has specific capability set:

- **Agent Management System (AMS)** exerts supervisory control over access to and use of MAGE platform. It is the managing center of MAGE platform.
- **Directory Facilitator (DF)** provides yellow pages services to other agents, such as service registration, searching and updating
- **Message Transport Service (MTS)** is the default communication method between agents on different agent platforms.
- **Agent** is the fundamental actor in MAGE which combines one or more service capabilities into a unified and integrated execution model. Each agent can be designed with particular function for different applications.

Software describes all non-agent, executable collections of instructions accessible through an agent. Non-agent software can be encapsulated with agent.

In addition, two auxiliary modules are provided to support designing agent systems: Agent Library and Function Components. User can compose different agents by using many kinds of function components provided by MAGE. Also user can select suitable agent class from agent library.

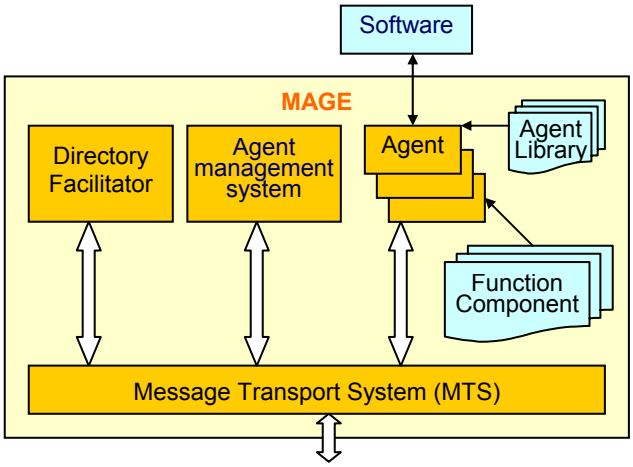


Fig 2 Architecture of MAGE platform

3.4 MAGE agent architecture

An agent in MAGE consists of six components: agent kernel, basic capabilities, sensor, communicator, function modules and knowledge base. Fig 3 demonstrates the detailed agent architecture.

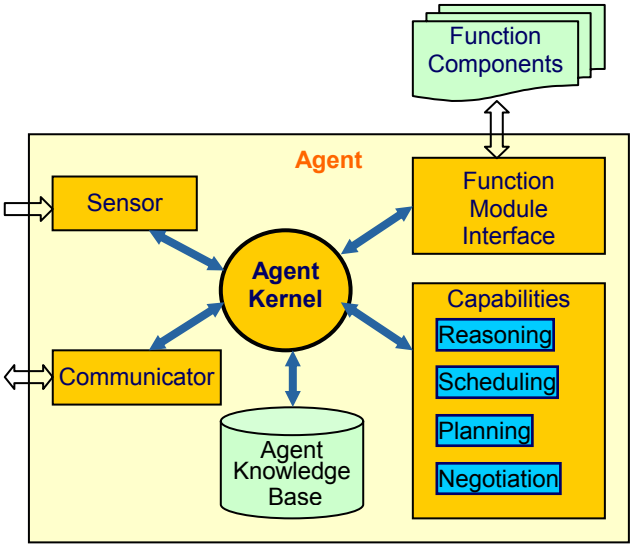


Fig 3 Architecture of MAGE agent

3.5 Agent life cycle in MAGE

Every agent in MAGE has a basic life cycle which includes six states: *Initiated*, *Active*, *Waiting*, *Suspended*, *Transit* and *Unknown*. And also MAGE support transition between different states, such as *Invoke* transform an agent from *Initiated* state to *Active* state. Refer to Fig 4 for detailed agent life cycle.

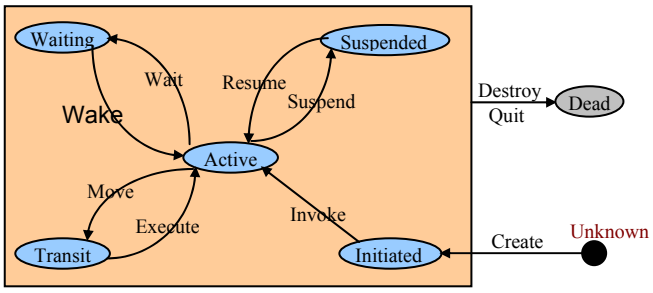


Fig 4 Agent life cycle in MAGE

3.6 Features of MAGE

3.6.1 Distributed computing platform

Distributed computing allows application program elements running on different hardware platform. MAGE is a distributed computing platform, and may be distributed on different host. MAGE is built on JAVA RMI and hides all implementation details from users, thus presents a unified computing environment for users. MAGE can support building distributed application easily.

3.6.2 Multiple schemes of software reuses

Software reuse can speed software development, reduce time to market and save resources, and so MAGE provides three kinds of method to reuse software (agent software or non-agent software coded in any language):

- **Embedded**
- **Executes from outside the system**
- **Dynamic link library (DLL)**

Through these methods, MAGE can provide software reuse in different granularity:

- **Reuse of application systems**
- **Reuse of sub-systems**
- **Reuse of components**
- **Reuse of functions***Multiple methods of agent generation*

MAGE provides three methods of agent generation:

- **Directly extends basic Agent class of MAGE**

This method aims at building new applications from MAGE.

- **Agent Description Language (ADL)**

ADL is used to describe the attributes (name, address, capabilities, etc) of an agent, and then MAGE can generate an agent according to this information and this agent has the corresponding attributes. This method aims at reusing software.

- **Clone**This is a way of agent generation at run-time.

3.6.4 Agent-oriented software design with a GUI

In specifying an agent system, we have found that it is highly desirable to adopt an external viewpoint. The description of an agent system from the external viewpoint is captured in two models:

- **Agent Model** describes the hierarchical relationship among different abstract and concrete agent classes and identities the agent instances which may exist within the system their multiplicity and when they come into existence.

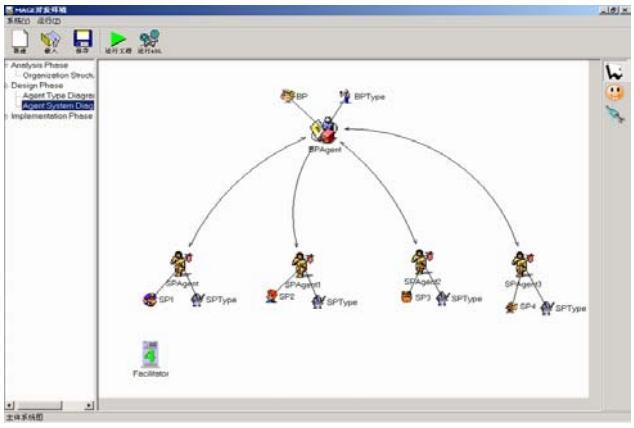


Fig 5 GUI of XMIDE

- **Interaction Model** describes the responsibilities of an agent class: the services it provides associated interactions

and control relationships between agent classes. This includes the syntax and semantics of messages used for inter-agent communication and communication between agents and other system components: such as user interfaces.

According to the two models, MAGE provides a graphic tool for agent-oriented software design: XMIDE, it is shown in Fig 5.

3.6.5 Open and FIPA compliant

The Foundation for Intelligent Physical Agents (FIPA) is an international organization that is dedicated to promoting the industry of intelligent agents by openly developing specifications supporting interoperability among agents and agent-based applications. MAGE is not isolated, it is open and FIPA compliant, so it can communicate with any agent system complying with FIPA.

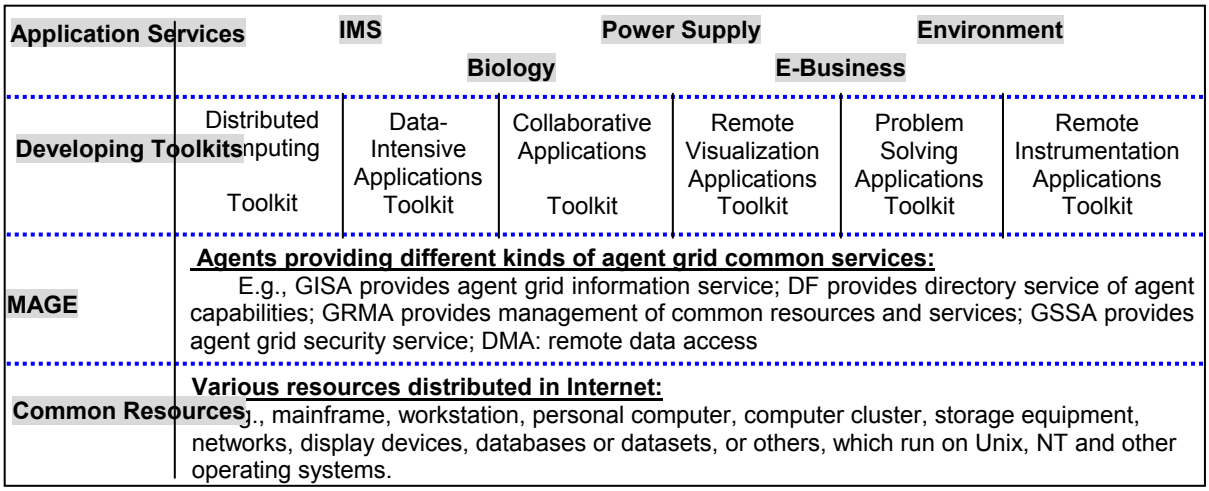


Fig 6 Architecture of AGEGC

Agentcities is a worldwide initiative designed to help realize the commercial and research potential of agent based applications by constructing a worldwide, open network of platforms hosting diverse agent based services. The ultimate aim is to enable the dynamic, intelligent and autonomous composition of services to achieve user and business goals, thereby creating compound services to address changing needs. Up to now (2002-10-9) there are sixty agent platforms connected to agentcities.net. Our MAGE platform connected with agentcities successfully in 2002-4, and is the only agent platform connected with agentcities in China, and also become the second agent platform in Asia that connected with agentcities.

4. AGENT-BASED GRID COMPUTING SYSTEM AGEGC

4.1 The Architecture of AGEGC

From the four-layer grid model discussed in Section 2, we can see that how to provide an Agent Environment in the second layer is the key in implementing a prototype of this agent grid model because of the following reasons:

- First, Agent Environment *integrates* the components of Common Resources and makes these resources thus available and useful.
- Second, Agent Environment *provides* different kinds of agent grid common services for upper layers, thus upper layers can make use of Common Resources as easily as possible.

So we need a powerful agent environment that is competent for this. As we can see from Section 3, MAGE is a distributed agent environment and has many advantageous features. Based on MAGE, we have implemented a prototype of agent gird: AGEGC. Fig 6 demonstrates the architecture of AGEGC.

In the following sections, several key issues of AGEGC will be discussed.

4.2 Directory Service

Grid applications often involve large amounts of data and/or computing and are not easily handled by today's Internet and web infrastructures. Grid technologies enable large-scale sharing of resources within groups of individuals and/or institutions. In these settings, the discovery, characterization, and monitoring of resources, services, and computations are challenging problems due to the considerable diversity, large numbers, dynamic behavior, and geographical distribution on the entities in which a user might be interested.

Consequently, directory services are a vital part of any grid software or infrastructure, providing fundamental mechanisms for discovery and monitoring, and hence for planning and adapting application behavior. In AGEGC, there are two types of directory service. Correspondingly, there are two types of agent: **DF** agent and **GISA** agent.

As we can see from the architecture of MAGE, **DF** (Directory Facilitator) is a mandatory component that provides a yellow pages directory service to agents. It is the trusted, benign custodian of the agent directory. MAGE may support any number of DFs and DFs may register with each other to form federations.

Every agent that wishes to publicize its services to other agents, should *register* its service description with DF. Also an agent can *deregister* itself form DF, which has the consequence that there is no longer a commitment on behalf of the DF to broker information relating to this agent. At any time, and for any reason, the agent may request the DF to *modify* its service description. An agent may *search* in order to request information from a DF.

We extend DF with a more abstract interface so that you can query: what agent A can do? Which agent is competent for a specific work? We have a reasoning machine embedded in DF, so it can "think"; if it thinks that a single agent can not do a specific job, it maybe return more than one agent whom together can do that job. And moreover, if it can't find agents that are capable, it may resort to other DFs.

GISA (Grid Information Service Agent) contains static and dynamic information about compute resources, as well as static and dynamic information about the network performance

between compute resources. It provides information directory service. You query GISA to discover the properties of the machines, computers and networks or other common resources that you want to use: What is the state of the computational grid? What resources are available? How many processors are available at this moment? What bandwidth is provided? Is the storage on tape or disk? GISA provides middleware information in a common interface to put a unifying picture on top of disparate equipment.
The GISA uses the Lightweight Directory Access Protocol (LDAP) as a uniform interface to such information.

4.3 Resources Management

GRMA (Grid Resource Management Agent) is an important agent that provides capabilities to do remote-submission job start up. GRMA unites Common Resources and services, providing a common user interface so that you can finish a job with any common resource or service. GRMA is a general, ubiquitous service, with specific application toolkit commands built on top of it.
The GRMA processes the requests for resources for remote application execution, allocates the required resources, and manages the active jobs. It also returns updated information regarding the capabilities and availability of the computing resources to GISA and DF.
GRMA provides an API for submitting and canceling a job request, as well as checking the status of a submitted job. We extend Globus Resource Specification Language (RSL) to describe request. Users write request in ERSL and then request is processed by GRAM as part of the job request.
For example, if one wants to start a MAGE platform, but his own machine is busy. Then he first queries DF which agent has the capabilities to provide information services. Then DF tells him that GISA agent can do it. Then he queries GISA which machine is free in ICS-DOMAIN with a command *GISA-query-machine –lim ICS-DOMAIN*. After he gets an answer that *junez.ics.ict.ac.cn* is free, he can start a MAGE platform on host *junez.ics.ict.ac.cn* with a request *AGEGC-run junez.ics.ict.ac.cn "java mage.Boot –gui"*.

4.4 Data Management

In an increasing number of scientific disciplines, large data collections are emerging as important community resources. In domains as diverse as global climate change, high energy physics, and computational genomics, the volume of interesting data is already measured in terabytes and will soon total petabytes. The communities of researchers that need to access and analyze this data (often using sophisticated and computationally expensive techniques) are often large and are almost always geographically distributed, as are the computing and storage resources that these communities rely upon to store and analyze their data.
DMA (Data Management Agent) is an agent mainly aiming at remote data access. DMA provides basic access to remote data. Operations supported include remote read, remote write and append.

5. APPLICATION EXAMPLES

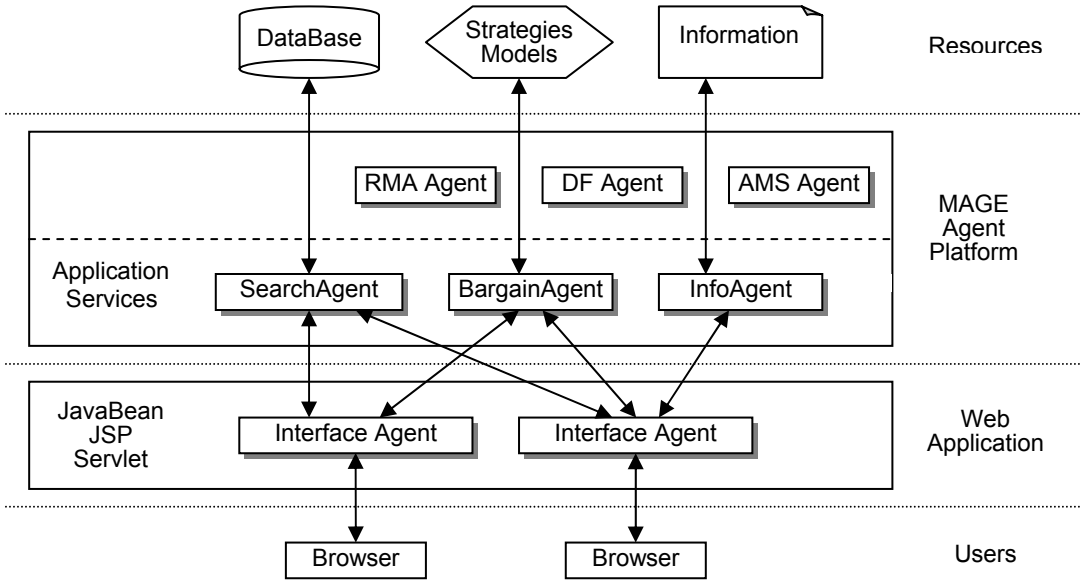
5.1 e-Business

We have developed a prototype of e-business. Figure 7 shows the architecture of agent-based e-business system. Four layers are constructed and each layer has its isolated functions.
The first layer includes all kinds of basic resources, such as database, models, and so on. Those resources are public and can be accessed by agent. For example, all data about the storage of e-business system are stored in a database, which is managed by a kind of DBMS such as Oracle or SQL Server. The database can be accessed by applications (agent) through standard SQL sentence.
The second layer is MAGE agent platform, which is the core of the system. It can be divided into two sub-layers, agent platform management and application services. Agent platform management is bases of the MAGE platform. It is in charge of the management of agent platform and management of all agents' life cycle. AMS, DF and RMA are included in this sub-layer. Another sub layer is application services. It is constructed by many kinds of agents and each agent has it particular functions for given application, such as SearchAgent and BargainAgent. SearchAgent provides the service of database access. BargainAgent can employ different strategies to bargain with other buyer agent, so it can act as seller agent.
The third layer lies in Web server and it provides many kinds of Web application. It bases on the Web and establishes a friendly interface between users and application systems. In this layer, many services are provided through web pages and can be implemented by JavaBean, JSP, Servlet, and so on.
When a user login this web, an interface agent will be generated corresponding to the user. Also the interface agent will register in MAGE platform .If the user wants to search for some goods, the query information will be generated by the interface agent and be sent to the SearchAgent. After process the query, SearchAgent will send the search results to the interface agent. Finally, the user can get the results from web page which is generated by the interface agent.
Another interface agent will be generated if the user wants to buy something on the web. It acts as buyer agent. The user can setup the bargain strategies and models freely.

5.2 Oil Supply Chain

Military logistic system is one of the typical applications of multi-agent system technology. In this section, we present an Oil Supply Chain System (OSCS), which was developed with MAGE. In this prototype system, OSCS simply includes six agents, shown in Figure 8
LandUnitAgent is a consumer of oil, while other five member agents, with cooperation, are collectively responsible for providing oil to the consumer agent. In more details, LangOilAgent and NavyOilAgent are specific agents designed for Land Oil Base and Navy Oil Base of military respectively. LogisticHeadAgent is in charge of scheduling among oil consumer, oil providers, (i.e., Navy Oil Base and Lang Oil Base) and Oil Refinery, which is acted by OilRefineryAgent. In the agent-based OSCS, TransporterAgent is delegated to transporter department, which is responsible for transporting oil from the oil bases to the oil consumer.
The OSCS agents are autonomous in the sense that they can perform different ACL acts to different outside agents in different contexts. For example, with the oil decreasing, LandUniAgent would send a REQUEST (ACL Message) to LandOilAgent for oil supply, when the oil storage goes under a pre-defined critical line. On receiving REQUEST message from LandUnitAgent, LandOilAgent would act variously, depending upon its current capability of oil. For example, if LandOilAgent can supply oil with the request volume, it would then send an AGREE message to TransporterAgent in order to inform the real-world Transporter Dep. to transport the oil requested by LandUnitAgent. Otherwise, LandOilAgent would reply a REFUSE message to LandUnitAgent and initiate a REQUEST to LogisticHeadAgent for oil supply too.

Fig 7 E-Business



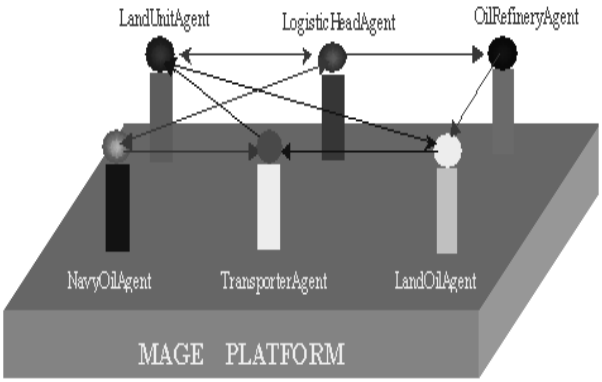


Fig. 8 Oil Supply Chain System

Though autonomous agents can execute designed tasks with their own mental states and behavior rules, people can also access them by specific GUIs. Specifically, OSCS was provided with user interfaces by which people can view and control agents' lifecycle states, browse the transaction records, modify agents' behavior rules and so forth.

To sum up, the agent-based OSCS system can efficiently help human to manage the real-world OSCS system. In particular, intelligent agents can shorten the delay in practice, and ensure that the oil supply chain can work smoothly in the varying environment.

6. CONCLUSIONS

Establishing grids is an important undertaking in developing scalable infrastructures. In this paper we have proposed a model for agent-based grid computing from the implement point of view. Based on the model agent-based grid computing system AGEGC has constructed by MAGE, which is a multiagent environment platform. AGEGC has applied to develop several application systems presented in the paper.

Due to the very generic nature of the grid computing, we can involve the research on it from different level, such as operating system layer, information layer, knowledge layer, service-oriented layer. Agent-based grid computing system AGEGC focuses on service-oriented layer in terms of current exist running environment. AGEGC will be useful platform for research on semantic grid.

7. REFERENCES

[1] Grimshaw, W. Wulf et al., The Legion Vision of a Worldwide Virtual Computer . Communications of the ACM, 40(1), 1997.

[2] Catlett and L. Smarr, Metacomputing, Communications of the ACM, June 1992, pp. 44-52.

[3] David De Roure, Mark A. Baker, Nick R. Jennings and Nigel R. Shadbolt, The Evolution of the Grid.

[4] Foster and C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit. International Journal of Supercomputer Applications, 11(2): 115-128, 1997.

[5] Foster and C. Kesselman (eds.), The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999.

[6] Keith G. Jeffery, Knowledge, Information and Data, A briefing to the Office of Science and Technology, UK, February 2000.

[7] Zhongzhi Shi, Intelligent Agent and Applications. Science press, China, 2000

A Global Optimisation Technique For Optical Thin Film Design

D.G. Li and A.C. Watson
School of Computer and Information Science
Faculty of Communications, Health and Science
Edith Cowan University
2 Bradford Street, Mount Lawley, W.A. 6050, Australia
E-mail: d.li@cowan.edu.au

ABSTRACT

Many advanced local and global optimisation techniques, such as Gradient, Simplex, Flip-flop, Needle , Genetic and Simulated annealing, have been successfully applied to optical thin-film design. Any optimisation algorithm applied to a particular design problem should firstly address the issue of choosing a reasonable starting design, which is always a big obstacle to an inexperienced designer. To find the true global optimised solution for a thin film design problem, we need to solve an array of interlinked multi-dimensional simultaneous equations. For more than just a few layers, until recently this has been a very difficult task, requiring the use of a supercomputer and highly skilled programming. By using the orthogonal Latin Square theory and an experimental design methodology in a search space reduction process, a Windows based program has been written that can operate on even a desktop personal computer. It can find the global optimum design for a 23 layers design using any dispersive and lossy material within a period of several hours. Additionally, this methodology (DGL-Optimisation, DGL is the short for D.G. Li) allows the use of target spectra such as s & p polarisation, with reflection and transmission simultaneously.

1. INTRODUCTION

The majority of problems faced by designers and engineers can be described as some form of local optimisation, trading off the improvement in one aspect against the worsening of another. In the thin film design problem, we need to find how to stack optical thin film layers on top of one other to obtain the optimum spectral response - changing one layer thickness may improve the performance of the stack at one wavelength, but worsen it at another.

For a particular target spectral response (reflectance and / or transmittance), the variables not only include the refractive index of each layer and its thickness, but also weighting factors for each wavelength, the material loss and dispersion, the ray's polarisation and angles, and for non-parallel layers, it might even include the layer's x and y coordinates.

To find the solution that best matches the target is a very difficult task (especially when there is more than 1 condition to be satisfied, such as both reflection and transmission s and p polarisation), using dispersive and lossy materials. The merit function plot would appear as a multi-peak, multi-variable plot. Because there are an enormous number of inter-related possible layer combinations, the best film design cannot be found by any simple analytical process. The methods currently used in thin-film design software, such as the damped least squares, simplex, needle, flip-flop optimisations and annealing algorithm, etc., all depend on a starting condition either selected by the user or generated internally by the program. These starting conditions may be completely hidden from the user. Changing the initial conditions will give a different result, and the user has no way of knowing how much improvement could be effected.

With the DGL-global optimisation algorithm embodied in OpTeFilm software, the user knows that the design found is optimised within the criteria set - there is no need to try other starting conditions for the same layer structure, because there are no starting guesses. The algorithm inexorably must find the optimum solution that exists within the boundary conditions. This efficiency has powerful economic consequences. For example, previous designs needing excessive numbers of layers can now be fabricated with fewer layers, lowering cost, to get the same performance and better yields. A manufacturer can improve yields on marginal designs by using a design with a greater margin of error, as well as offering previously unavailable products.

2. THE MERIT FUNCTION

As in other methods, a merit function (M.F.) is calculated for each design the program finds. The M.F. is calculated by the sum over all the wavelengths (or angles) of a difference function between the value of the parameters calculated by the current design compared to the target. The optimisation process then becomes the searching for the design that has the best merit function. Thus as an example, for a target having a reflection spectra for both s and p polarisation, the merit function would be calculated as :

$$MF = \sum_{\lambda} \{ | Rp - \mathrm{Rpo} | + | Rs - \mathrm{Rso} | \}$$

where:

MF	Calculated Merit Function -	current design
s & p	Polarisation State	
R_{so} R_{po}	Target spectra -	Specified by the user
R_s R_p	Transmittance or reflectance -	current design

Therefore by finding the lowest value of MF we will have the best average design.

This formula can be made more sophisticated by introducing weighting factors to increase the importance of user specified wavelengths, as well as using other forms of the difference between the target and current design such as a square. A square form has the effect of weighting differences which are greater, thus flattening the deviation between the target and design.

3. A GRAPHICAL ANALOGY OF OPTIMISATION METHODS

As discussed earlier, a single number (the M.F.) can be used to describe how close to the target a current design is located. By plotting a multi-dimensional graph with merit function as one of the axes, we can visualise the process. We require as many orthogonal axes as the number of variables plus one for the M.F. Thus, for a 2 layer thin film problem, we require a 3-dimensional plot.

To see the process used in a simplified form, image a 2-dimensional array along the x and y axes, which corresponds to a 2 layer problem. Let the value along the x axis represent a thickness of layer 1, and along the y axis the thickness of layer 2. In the z direction we plot the Merit Function value.
The task of the program is then to find the x and y values that generate the lowest value of the M.F. In this description, we shall invert the M.F. since a peak is easier to see than a valley. Therefore we choose the form of M.F. whose value increases as the design performance improves, i.e. we want to maximise the M.F. (One form of M. F. can be transformed into the other by taking a reciprocal of it). The game is to change each layer thickness in order to maximise the Merit Function. To help understand the optimisation process, consider the analogy of a man wandering in a cratered terrain, with a Global Positioning Satellite (GPS) receiver, which displays his absolute x, y, and z coordinates. Height (z) is the Merit function, and x, y represent the layer thicknesses of each layer in a 2 layer design. His task is to find the highest point (largest M.F.) bounded by the user defined maximum and minimum values of x and y . See Fig 1a.

3.1 Conventional methods

If he just walks upwards until he can go no further uphill, he will have found the local maximum. The 'best guess approach' is based on a starting design (position) that may be based on many years of experience. There are several mathematical methods available such as the gradient method, which alter several layer thicknesses simultaneously, see what happens to the merit function, and move the design in the direction of a maximum of the Merit Function. These find the 'Local Optimum, and the end design is completely dependent on the starting guess.
A derivation of this is the Simplex Method which, by the use of triangulation and in some variations, random numbers, is able to find a better 'near local' maximum quicker, exploring other designs by 'jumping away' from the nearest peak.
For multi-layer designs, using only two different materials, the flip-flop[1,2] method uses a method based on a large number of alternating sub-layers. Briefly, the merit function of the stack as each sub layer is flipped from low to high or vice versa is calculated. Stack sequences which improve the merit function when flipped, are preserved, if it was worsened, then the previous state is preserved. The program flips each sub-layer in turn, and the process is repeated when the first round is completed. By this process the number of layers is progressively reduced as adjacent high layers are coalesced. Similarly for low index layers. The process continues until the number of layers is below that specified by the user or no further improvement can be made.
The needle[3] optimisation method uses a completely different technique to optimise the design, by adding layers one by one, optimising the combination each time, and then adding another layer, and so on. This has the apparent advantage of keeping the layer number to a minimum for an acceptable design. It does have the advantage of being capable of producing complex designs with a minimum of user interaction, however the design found is unlikely to be the global optimum. The final design however is very dependent on the initial starting thickness, and many more layers than necessary are often required for a given performance.

3.2 The DGL method

The DGL-optimisation operates by a process of searching for all regions in the layer thickness space where a height greater than a specific level is located. This is akin to creating a contour map by slicing parameter space at a constant value of merit function.
In our 2 layer analogy, this is the equivalent of a plane parallel to the x-y plane at height z. See Fig 1b. This plane intersects the topography and identifies the entire region within which the peak is known to lie. By raising the slicing plane repeatedly, the region within which the peak must lie, is made smaller and smaller until only the highest peak remains - See Fig. 1c. Its coordinates correspond to the layer thicknesses of the optimum design. In practice, the surface is a mathematical construct of as many orthogonal dimensions as there are layers!
No starting guess is necessary (or even possible), and the operator only has to define the basic parameters, such as the number of layers, the max. and min. layer thicknesses for each layer (i.e. the boundary conditions), materials, and target spectrum. After the program is started, the operator can observe the values of max. and min. (within which the global optimum resides) for the various layers approaching each other. At the end of the run, there will be no layer whose max. and min. value is greater than the specified value (as little as 1 nm). This stopping value can be thought of as being the dimension of the peak - if one wishes one can make this as small as one would wish, in practice it just takes longer but with no practical benefit.
By using DGL global optimisation functions, a designer can be assured that he has found the best design physically possible, independent of his so-called best guess.
The mathematical procedures used in this form of global optimisation are possible to apply to a variety of other previously unsolved problems relating to the resultant of dependent variables, including experimental design and manufacturing variations. In the visible region, one of the global optimisation functions of OpTeFilm will even find all different designs having the same color, and compare them for manufacturing variations.
There are many other approaches people have adopted, but until now (with the exception of scanning), they all depend either on a starting design, some form of local optimisation or some random variation. Each method will usually give rise to different solutions. For designs using a large number of layers, these are still the only methods possible. In contrast, the DGL optimisation described here is a methodical global method.

4. PRINCIPLES OF THE DGL GLOBAL OPTIMISATION AND THE LATIN SQUARE

The true magnitude of the problem can be seen by considering a scanning approach, i.e. measuring the merit function value for every possible combination of layer thicknesses. The scanning method is guaranteed to find the global optimum, provided one does enough calculations. For example, let us consider the number of merit function calculations necessary to scan a 5 layer design at 2nm intervals over 10 nm to 350 nm (170 measurements per layer). The total number of possible combinations is 170^5 , which would take years for even a super-computer!
The DGL optimisation uses a mathematical method based on orthogonal sets of numbers. By slicing the multi-dimensional parameter space with a horizontal plane of the Merit Function, with each parameter independent of the others. A peak is always be surrounded by a slope. By finding all regions in which the merit function has values above that of the plane, one can narrow the search region. After finding the boundary of all the isolated regions where this occurs, the plane is raised again, and the process repeated.
A Latin Square is an array of numbers widely used in experimental designs, By utilising orthogonal Latin Squares, one can form an array of several dimensions which are

orthogonal to each other, and therefore allow the calculation of a resultant using many interdependent variables. This allows us to zoom in on the area of interest. Each iteration reduces the boundary of 1 or more of the variables in parameter space until the boundary dimension is less than that specified by the stop criteria. Combining Latin Square sampling with function domain contraction techniques, results in an optimisation with two desirable properties. Firstly, the number of function evaluations can be greatly reduced, and secondly, there is a guarantee of finding the global optimum solution. There is a text available on Latin Square theory and its applications[7].

5. MATHEMATICAL FORM OF DGL OPTIMISATION

Consider a multi-dimensional continuous function $f(\boldsymbol{x})$ with multiple global minima and local minima on subset G of R^n

(i) We define the local minima as follows
For a given point $\boldsymbol{x}^* \in G$, if there exists a δ-neighborhood of $\boldsymbol{x}^*$, $O(\boldsymbol{x}^*, \delta)$, such that for
$x \in O(\boldsymbol{x}^*, \delta)$,
and $f(\boldsymbol{x}^*) \le f(\boldsymbol{x})$ (1)
then $\boldsymbol{x}^*$ is called a local minimal point of $f(\boldsymbol{x})$.
(ii) Definition of global minima
If for every $\boldsymbol{x} \in G$ the inequality (1) is correct, then $\boldsymbol{x}^*$ is called a global minimum of $f(\boldsymbol{x})$ on G, and the global minima of $f(\boldsymbol{x})$ on G form a global minimum set.
(iii) How to find the global minima
Now for a given constant C_0 such that the level set
$H_0 = \{\boldsymbol{x} | f(\boldsymbol{x}) < C_0, \boldsymbol{x} \in G\}$ is non-empty,
if $\mu(H_0)=0$, where μ is the Lebesque measure of H_0,
then C_0 is the minimum of $f(\boldsymbol{x})$ and H_0 is the global minimum set.
Otherwise, assume that $\mu(H_0)>0$ and C_1 is the mean value of $f(\boldsymbol{x})$ on H_0.
Then $C_1 = 1/\mu(H_0) \int_{H_0} f(x) \mathrm{d}\mu$ (2)
and
$C_0 \ge C_1 \ge f(\boldsymbol{x}^*)$ (3)
We then gradually construct the level set H_k and mean value C_{k+1} of $f(\boldsymbol{x})$ on H_k as follows:
$H_k = \{\boldsymbol{x} | f(\boldsymbol{x}) < C_k, \boldsymbol{x} \in G\}$ (4)
and
$C_{k+1} = 1/\mu(H_k) \int_{Hk} f(\boldsymbol{x}) \mathrm{d}\mu$ (5)

With the assistance of Latin Square sampling, a decreasing sequence of mean values $\{C_k\}$ and a sequence of level sets $\{H_k\}$ are obtained.
Let $\lim_{k \to \infty} C_k = C^*$ (6)
and
$\lim_{k \to \infty} H_k = H^*$ (7)

It can be proven that C^* is the minimum of $f(\boldsymbol{x})$ on G, and H^* is the global minimum set.

There are several strategies to avoid missing the global optimum when seeking the minimum solution. Among these, the most important step is to design a suitable orthogonal Latin Square with which the function within domains can be repeatedly sampled. The algorithm is automatically constrained to stay within the function domain and will not request function evaluations outside this domain.
There are two stopping criteria possible; either when the target M.F. value is reached, or when the maximum domain length is smaller than the user selected value. OpTeFilm uses the latter stop criteria, corresponding to the variation possible for each layer thickness - which can be as little as 1 nm. This means that the global minimum has been found for a particular layer thickness range of each layer, with a variation of less than 1 nm for each layer, strictly speaking then, the global optimum is not defined at a point but as lying within a region.

6. APPLICATION OF DGL GLOBAL OPTIMISATION TO PRACTICAL FILTER DESIGN

In figs 1a, 1b & 1c, we see a representation of a two layer problem. Using a search analogy, to find the peak, the region in which the highest peak must lie is narrowed each time the plane is raised. This occurs until there is only one peak left. Its coordinates correspond to the layer thicknesses of the optimum design.
The examples in this paper are intended only for the comparison of solutions obtained with the DGL method with those obtained with other thin film synthesis programs. Dispersion of the refractive indices and residual absorption within the dielectric layers have not been allowed for since they do not materially affect the solution. Any refractive indices lying within reasonable upper and lower limits were accepted.
OpTeFilm is the first true global optimisation program to run under windows on a PC. The parameters that the user defines must include the materials (with specified refractive index and loss over the wavelength range), the number and order in which the layers are to be placed, the incident angle etc. The target can be selected form a variety of pre-selected parameters, including *s* and *p* polarisation. The advantage of this methodology is that a global optimisation can be made on parameters that may or may not be related. For example a merit function can be calculated including *s* and *p* polarisation of transmission and just *p* reflection. A weighting factor can also be applied to specific target wavelengths.
Figure 2 shows a typical design process for an edge filter. Starting from a theoretical design, the structure can be optimised locally by the gradient method. A Simplex optimisation improves the design considerably. This would often be the endpoint. Using a global optimisation however, with the same number of layers and no starting guess, yields the best performance. Note must be made that a different performance can arise if the order of the materials is reversed. i.e. the high index next to the substrate is substituted by a low index material.
Figure 3 shows a comparison between the best design found in a competition[4,5] for the best anti-reflection germanium I.R. filter. The global optimisation clearly yields a superior performance.
Figure 4 is a comparison of designs for a 50% beamsplitter. Here the comparison is between the Gradient[8] and the DGL optimisation.
The application of this technique will undoubtedly have implications well beyond thin film design. Already some of these principles have been applied to a variety of difficult mathematical problems involving image recognition, hydrology and lens design, as well as the theoretical solving of mathematical problems by evolutionary optimisation[6].

7. CONCLUSION

By using a Latin Square and other mathematical techniques, it is possible to create a global optimisation program for thin film design on the desktop. A global optimisation program such as OpTeFilm allows the solution of multilayer designs using

mixed parameters such as separate *p* and *s* transmission spectra, weighted targets, using lossy and dispersive materials without any starting design. Using the criteria selected by the user, the function will methodically proceed to the optimum design. The primary advantages of this technique are that mixed targets with dispersive and lossy materials can be used, the global optimum is always found, excellent designs can be found with little prior knowledge, and the new merit functions can be created according to whatever combination of parameters are required.

8. REFERENCES

[1] J.A. Dobrowolski. "Comparison of the Fourier transform
[2] and flip-flop thin-film synthesis methods," Applied
[3] Optics **25**, 1966 (1986)
[4] J.A. Dobrowolski and R.A. Kemp. "Flip-flop thin film design program with enhanced capabilities," Applied Optics **31** 3807 (1992)
[5] A.V. Tikhonravov. "Some theoretical aspects of thin-film optics and their applications," Applied Optics **32** 5417 (1993)
[6] J.A. Aguilera et al. "Antireflection coatings for germanium IR optics: a comparison of numerical design methods," Applied Optics **27**, 2832 (1988)
[7] P. Baumeister. "Starting designs for the computer optimization of optical coatings," Applied Optics **34**, 4835 (1995)
[8] H. Bersini et al. "Results of the first intenational contest on evolutionary optimisation," IEEE journal, (1996)
[9] W.G. Cochran and G.M. Cox, "Experimental designs", John Wiley & Sons, Inc, (1957)
[10] J.A. Dobrowolski and R.A. Kemp, "Refinement of optical multilayer systems with different optimization procedures", Applied optics **29** 2876 (1990)

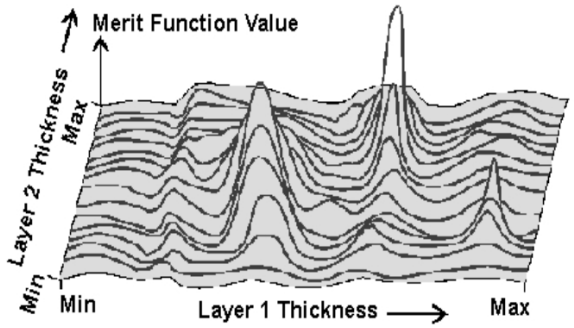


Fig. 1(a) Merit Function for 2 layer Problem

In this description of a 2 layer design, the boundary conditions are defined by the layer's maximum and minimum thicknesses. We seek to find the layer thicknesses that give rise to the maximum value of the Merit Function.

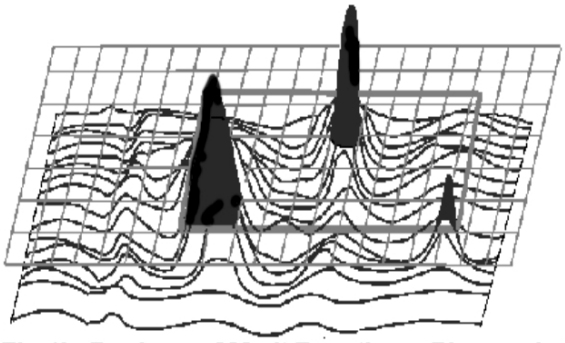


Fig. 1(b) Regions of Merit Function>Plane value

A plane is constructed of 'constant Merit Function' and the boundary of regions having a higher merit function than that of the plane are identified.

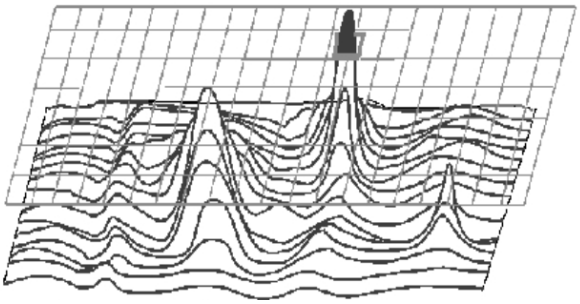


Fig. 1(c)Merit Function Peak region

As the plane is raised, the region within which the peak of the Merit Function exists, is narrowed. The process is repeated until the layer thicknesses which give rise to the highest peak are uniquely identified.

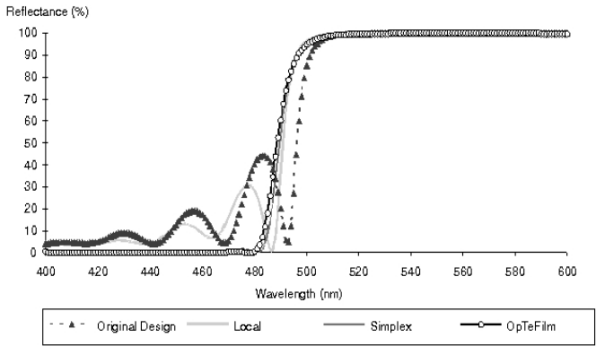


Fig. 2(a) Edge Filter Performance-Detail

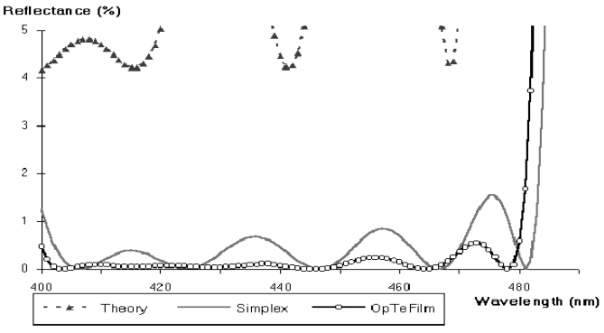


Fig. 2(b) Edge Filter Performance-Detail

A comparison of the results of using different optimisation

processes. The global optimum design has a significantly lower average reflectance. All designs use the same number of layers.

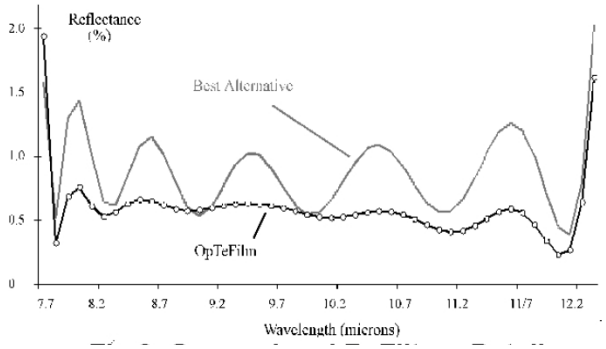


Fig. 3 Germanium 1.R. Filter-Detail

As above, the starting criteria selected for the OpTeFilm design consisted of the same as that of the published design, viz. Number of layers and their order, materials, etc.

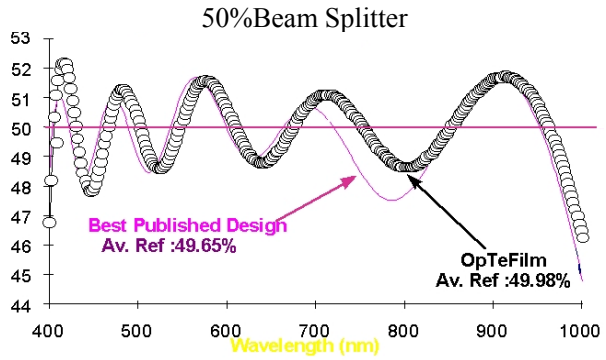


Fig. 4 50% Beamsplitter - The starting criteria selected for the OpTeFilm design consisted of the same as that of published design obtained by Gradient optimisation, viz. Number of layers and their order, materials, etc

Modelling the Runtime of the Large and Sparse Linear System Solver on Parallel Computers*

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

ABSTRACT

For the solutions of large and sparse linear systems of equations with nonsymmetric coefficient matrices, we propose the IQMR method 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 and matrix-vector multiplications 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 show the execution time of collective communication operations in PVM and MPI can be modeled by runtime functions in closed form on CrayT3D and CrayT3E. We also demonstrate that the runtime functions can be used to model the computation and communication of the IQMR method.

Keywords: Large and sparse linear systems, iterative methods, the IQMR method, Lanczos process, PVM, MPI, global communication, performance model

1. INTRODUCTION

One of the fundamental task of numerical computing is the ability to solve linear systems with nonsymmetric coefficients. These systems arise very 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 subproblems in linear and nonlinear programming.

One of them, the quasi-minimal residual (QMR) method [8], uses the Lanczos process [7] with lookahead, a technique developed to prevent the process from breaking down in case of numerical instabilities, and in addition imposes a quasi-minimization principle. This combination leads to a quite efficient algorithm among the most frequently and successfully used iterative methods. This method are widely used for very large and sparse problems, which in turn are often solved on massively parallel computers.

Recently, we have proposed a new improved twoterm recurrences Lanczos process [13] without lookahead as the underlying process of QMR. The resulting algorithm is reorganized without changing the numerical stability so that all inner products and matrixvector multiplications of a single iteration step are independent and communication time required for inner product can be overlapped efficiently with computation time. Therefore, the cost of global communication which represents the main bottleneck of parallel performance on parallel distributed memory computers [11, 12, 16] can be significantly reduced. The improved IQMR (IQMR) algorithm maintains the favorable properties of the original algorithm while not increasing computational costs.

*The author's Email is: lyang@stfx.caformulas

Now, MPI is probably the most commonly used message passing library for programming distributed memory parallel computers. Implementations of MPI are available for all commercially available parallel architectures and are generally accepted by application programmers. But to find an efficient parallel version of a scientific and engineering algorithm such as the IQMR method, still takes a lot of time for programming and measuring different versions of the algorithm. This is due to the fact that scientific and engineering algorithms on parallel computers show a complicated runtime behavior caused by the overheads of explicit communication or synchronization and load imbalance. In this article, we first investigate and compare singletransfer and collective communication operations both in isolation and in the context of large message passing application programs. We will then look at mainly the collective communication operations such as singlebroadcast which is mainly used for modelling the communication cost of inner products. For the operation we consider several variants being realized in different ways within the PVM or MPI environment. The runtime prediction provides functions in closed form in the spirit of the function for single-to-single transfer linear in the message size, which is valid for a large number of machines. The functions depend on several parameters including application as well as machine parameters. The values of the parameters are determined with the least-squares method. The resulting runtime formulas describe the communication time quite accurately. The results of the experiments and the modelling can be used in compiler tools, in parallelizing compilers or directly by the application programmer. A knowledge of the runtime behavior of such communication operations can have a great inuence on the design process of the efficient parallel IQMR method. The integration of the runtime formulas into a cost model allows us to predict the runtimes for the entire IQMR method quite accurately.

The paper is organized as follows. We will describe briefly the improved quasi-minimal residual (IQMR) method in section 2. In sections 3 and 4, the parallel systems and performance model are presented including the communication model and assumptions for computation time and communication costs, respectively. Particularly we will describe the modelling of execution time of collective communication operations in PVM and MPI on CrayT3D and Cray T3E parallel systems. The corresponding theoretical complexity analysis will be presented in section 4.3. Finally we offer some limited preliminary measured and estimated timing results, respectively.

2. THE IMPROVED QMR METHOD

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

$Ax=b$, where $A \in R^{n\times n}$ and $x,b \in R^{n}$. (1)

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

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

Here x_0 is any initial guess for the solution of linear systems, $r_0 = b\text{-}Ax_0$ is the initial residual, and $K_n(r_0;A) = \text{span}\{r_0; Ar_0; \cdots; A_{n-1}r_0\}$, is the n-th Krylov subspace with respect to r_0 and A.

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

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

where V_n is generated by the improved Lanczos process [15, 14, 11], which is derived in such a way that all matrix-vector multiplications, vector updates and inner products of a single iteration step are in parallel, and z_n is determined by a quasi-minimal residual property.

Algorithm 1 Improved Quasi-Minimal Resklunl Method

1: $\bar{v}_1 = \bar{w}_1 = r_0 = b - Ax_0, \lambda_1 = 1, \kappa_0 = 1, \mu_1 = 0,$

2: $p_0 = q_0 = u_0 = d_0 = f_0 = 0, \gamma_1^2 = (\bar{v}_1, \bar{v}_1), \xi_1^2 = (\bar{w}_1, \bar{w}_1)$

3: $s_1 = A^T \bar{w}, \rho_1 = (\bar{w}, \bar{v}_1), \varepsilon_1 = (s_1, \bar{v}_1), \tau_1 = \frac{\varepsilon_1}{\rho_1};$

4: for n =1,2,⋯,do

5: $q_n = \frac{1}{\xi_n} s_n - \frac{\gamma_n \mu_n}{\xi_n} q_{n-1};$

6: $\bar{w}_{n+1} = q_n - \frac{\tau_n}{\xi_n} \bar{w}_n;$

7: $s_{n+1} = A^T \bar{\omega}_{n+1};$

8: $t_n = A\bar{v}_n;$

9: $u_n = \frac{1}{\gamma_n} t_n - \mu_n u_{n-1};$

10: $\bar{v}_{n+1} = v_n - \frac{\tau_n}{\gamma_n} \bar{v}_n;$

11: $p_n = \frac{1}{\gamma_n} \bar{v}_n - \mu_n p_{n-1};$

12: if $(r_{n\text{-}1}, r_{n\text{-}1}) < \text{tol}$ then

13: quti

14: else

15: $\gamma_{n+1}^2 = (\bar{v}_{n+1}, \bar{v}_{n+1});$

16: $\xi_{n+1}^2 = (\bar{w}_{n+1}, \bar{w}_{n+1});$

17: $\rho_{n+1} = (\bar{w}_{n+1}, \bar{w}_{n+1});$

18: $\varepsilon_{n+1} = (s_{n+1}, \bar{v}_{n+1});$

19: $\mu_{n+1} = \frac{\gamma_n \xi_n \rho_{n+1}}{\gamma_{n+1} \tau_n \rho_n};$

20: $\tau_{n+1} = \frac{\varepsilon_{n+1}}{\rho_{n+1}} - \gamma_{n+1} \mu_{n+1};$

21: $\theta_n = \frac{r_n^2 (1 - \lambda_n)}{r_n r_n^2 + \gamma_{n+1}^2};$

22: $\kappa_n = \frac{-\gamma_n \tau_n \kappa_{n-1}}{\lambda_n r_n^2 + \gamma_{n+1}^2};$

23: $\lambda_n = \frac{\lambda_{n-1} r_{n-1}^2}{\lambda_{n-1} r_{n-1}^2 + \gamma_n^2};$

24: $d_n = \theta_n d_{n-1} + \kappa_n p_n;$

25: $f_n = \theta_n f_{n-1} + \kappa_n u_n;$

26: $x_n = x_{n-1} + d_n;$

27: $r_n = r_{n-1} - f_n;$

28: end if

29: end for

Based on the similar idea in [1, 2] with computation rearrangement, we derive an improved QMR method (IQMR) based on coupled two-term recurrences with scaling of both sequence of Lanczos vectors for maintaining the numerical stability. All inner products and matrix-vector multiplications of a single iteration step in the IQMR are independent and communication time required for inner product can be overlapped efficiently with computation time. The framework of this improved QMR method using the improved Lanczos algorithm based on two-term recurrences as underlying process is depicted in Algorithm 1.

The improved QMR method can be efficiently parallelized as follows:

- The inner products of a single iteration step (12), (15), (16), (17) and (18) are independent.
- The matrix-vector multiplications of a single iteration step (7) and (8)are independent.
- The communications required for the inner products (12), (15), (16), (17) and (18) can be overlapped with the update for pn in (11).

Therefore, the overhead of communication on parallel distributed memory computers can be significantly reduced.

3. PARALLEL SYSTEM OVERVIEW

The parallel computing systems we mainly use are the Cray T3D and T3E distributed memory multiprocessors (DMMs) with up to 2048 processing elements (PE) [4, 10]. The following detailed system description is taken mainly from [9]. For the T3D, each PE contains a DEC Alpha 21064 microprocessor (clocked at 150MHz) with an 8 KB instruction cache and an 8 KB data cache. One processing element node comprises two PEs and a network interface; the nodes are connected by a three-dimensional torus network. The network is composed of communication links and network routers which transfer packets through the communication links. The six communication links of each node are able to simultaneously support hardware transfer rates of 300 MB/s. The routers contain an X-dimension switch, a Y-dimension switch, and a Z-dimension switch which control the flow of packets through the different dimensions using the routing information of the packet. The X-dimension switch steers packets from one X-dimension communication link to the other or from one X-dimension communication link to the Y-dimension switch. The Y-dimension switch and the Z-dimension switch work identically. Although the memory is physically distributed, a logically shared memory is provided, i.e., any microprocessor can access data in the local memory of any PE without involving the microprocessor in that PE. Each virtual address is converted into a logical node number, PE number, and local address offset. If the PE number matches the PE number of the generating PE, a local memory access is performed. If the numbers do not match, a remote memory access is initiated by sending the PE number and the local address offset to the network interface. A remote memory read can be performed by the function *shmem get( )* from the Shared Memory Access (SMA) library [3]. This function copies a number of words from the local memory of a specified PE to the memory of the calling PE. A remote memory write can be performed using *shmem put( )* which copies a number of words from the memory of the calling PE into the memory of a specified PE. None of these calls changes the entries of the data cache of the remote processor. The startup time for both calls lies between 1 and 2 μs. The bandwidth for *shmem put( )* is 120 MB/s. The bandwidth for *shmem get( )* is 60 MB/s because a remote load consists of a request followed by a remote write by the remote PE to the calling PE. The T3D provides special hardware support for fast barrier and eureka synchronization. Both can be used if the entire user partition participates in the synchronization. Otherwise, software synchronization has to be used. The T3E uses an Alpha 21164 microprocessor clocked at 300 MHz, 450 MHz (T3E-900) or 600 MHz (T3E-1200) with an 8 KB direct-mapped instruction and data cache and an 96 KB three-way associative on-chip L2 cache. The 3D

interconnection network provides a bidirectional bandwidth of 600 MB/s for each physical link. The access to remote data is performed via 640 E-registers. The latency for an access to local and remote memory is 283 *ns* or 1500 *ns*, respectively. The sustained bandwidth to local and remote memory is 630 MB/s and 300 MB/s, respectively.

4. THE PERFORMANCE ANALYSIS

Based on these mathematical background described above, we will make the following assumptions suggested in [5, 6, 12, 16, 11] for our performance model. First, the model assumes perfect load balance and each processor holds a sufficiently large number of successive rows of the matrix, and the corresponding sections of the vectors involved. That is, our problems have a strong data locality. Secondly, we can compute the inner products (reduction) in two steps because the vectors are distributed over the processor topology. The computation of an inner product is executed by all processors simultaneously without communication or locally and these partial results are combined by a global sum operation and accumulated at a single destination processor, called *single-node accumulation* (SNA). The second phase consists of reversing the directions and sequence of messages sending the final results from this single processor to all other processors, called *single-node broadcast* (SNB).

In the following part we will describe a simple performance model including the computation time and communication cost for the main kernels as we presented before based on our assumptions. These two important terms are used in our paper suggested in [6]:

- Communication Cost: The term to indicate all the wall-clock time spent in communication, that is not overlapped with useful computation.
- Communication Time: The term to refer to the wall-clock time of the whole communication. In the non-overlapped communication, the communication time and the communication cost are the same term.

4.1 Computation time

The IQMR method contains three distinct computational tasks per iteration

- Two simultaneous matrix-vector products, $A\bar{v}_n$ and $A^T\bar{w}_{n+1}$ whose computation time are given $2t_{fl}\,N/P$.
- Five simultaneous inner products , $(\bar{v}_{n+1},\bar{v}_{n+1})$, $(\bar{w}_{n+1},\bar{w}_{n+1})$, $(\bar{w}_{n+1},\bar{v}_{n+1})$, $(\bar{s}_{n+1},\bar{v}_{n+1})$, and (r_{n-1},r_{n-1}) whose computation time are given by $(2n_{z-1})t_{fl}N/P$.
- Nine vector updates, q_n, $(\bar{w}_{n+1},u_n,\bar{v}_{n+1})$, p_n, d_n, f_n, x_n and r_n which are given $2t_{fl}N/P$.

where N/P is the local number of unknown of a processor, t_{fl} is the average time for a double precision floating point operation and n_z is the average number of non-zero elements per row of the matrix.

The complete (local) computation time for the IQMR method is given approximately by the following equation:

$$T_{comp}^{IQMR} = (19+2Nz)\frac{N}{P}tf_t = f(m)\frac{N}{P} \qquad (4)$$

4.2 Communication cost

Before we compute the communication cost for the IQMR method, we need to know the communication cost of the single-node broadcast. For the single-node (accumulation or) broadcast operation which send the same message from one processor to all other processors involved, the following variants in Table 1 are mainly considered in [9] for PVM. Based on the study in [9], considering the execution time of the broadcast operation on 256 processors of the T3D and T3E for different messages sizes shows that for small message sizes of up to 4096/8192 byte, all variants except the *pvm_psend* variant have nearly the same execution time. If we look at the larger message sizes, the piecewise operations from their study in [9] show the best performance which means that it is better to split a large broadcast message into small pieces than to send it as a large message. It is also concluded from their study in [9] that the difference is more than an order of magnitude for messages of more than 1 Megabyte. This is usually not expected for the distributed memory systems. The detailed explanation for this communication behavior can be found in [9]. A direct comparison between the *MPI_Bcast*() operation and the (piecewise) *PVM_broadcast* operations shows that the MPI operations are faster than the PVM operations [9].

Table 1 Realizations for single-node broadcast operation in PVM[9]

Send-operation	Receive-operation	Package mode
pvm_bcast	*pvm_rccv*	*PumDataDefault*
pvm_psend	*pvm_rccv*	*PumDataDefault*
pvm_bcast	*pvm_precv*	*PumDataInPlacet*
pvm_bcast	*pvm_rccv*	*PumDataInPlace*
piecewise *pvm_bcast*	*pvm_rccv*	*PumDataInPlace*
piecewise *pvm_bcast*	*pvm_precv*	*PumDataInPlace*

A comparison of the runtimes for maximum accumulation operations with PVM and MPI for 256 processors of the T3D with *pvm_reduce* and *MPI_Reduce*, respectively, shows that for small message sizes, the PVM operation is much faster than the MPI operation [9]. For larger message sizes, both operations have about the same runtime, but the MPI operation shows much more fluctuations than the PVM operation. For messages of more than 8000 bytes, the difference is usually below 10% [9].

Now, we will show how the runtimes of the collective communication operations on the T3D and T3E can be modelled by using runtime formulas. Here note that runtime formulas depend on various machine parameters including the number of processors, the bandwidth of the interconnecting network, and startup times for the corresponding operations. We present the modeling results for our work by using the fastest variants of PVM and MPI communication operations, respectively. The runtime behavior of the other variants can be modelled similarly. The corresponding PVM and MPI operations can be modelled by the same runtime function with different coefficients. The runtime function for single-node broadcast or accumulation are summarized as follows:

$$t_{snb}(P,b,V)=\tau(V)\log_2 P+\tau_c(V)b\log_2 P \qquad (5)$$

where the value b is the message sizes in bytes, P is the number of processors and V is the specific variant of the communication operation. The specific values for the coefficients $\tau(V)$, $\tau_c(V)$ are given in Table 2 [9]. The values result from curve fitting with the leastsquares method. The logarithmic dependence on the number P of processors is used because the broadcast transmissions are based on broadcast trees with logarithmic depth. The same formula can also be used for the prediction of single-accumulation operations.

The results have demonstrated the accuracy of the predicted and measured runtimes for PVM and MPI single-node accumulation operation. For the PVM broadcast, the

piecewise variant with the *InPlace* option is used.
Finally, based on the single-node accumulation and broadcast time for 1 inner product, the communication time of the IQMR method is given as follows:
TIQMR

$$T_{comm}^{IQMR} = 2\tau(V)\log_2 P + 2\tau_c(V)b\log_2 P \qquad (6)$$

4.3 Theoretical analysis

In this section, we will focus on the theoretical analysis of the parallel performance of the IQMR method where the efficiency, speed-up and runtime are expressed as functions of the number of processors scaled by the number of processors that gives the minimal runtime for the given problem size. The total runtime for the IQMR method is given by the following equation:

$$T_P^{IQMR} = T_{comp}^{IQMR} + T_{comm}^{IQMR} = f(m)\frac{N}{P} + g(m). \qquad (7)$$

This equation shows that for sufficiently large P the communication time will dominate the total runtime.
Let P_{max} denote as the number of processors that minimizes the total runtime Tp for any P processors of IQMR method, we can easily get

$$P_{\max} = \frac{f(m)N\ln 2}{2\tau(V) + 2\tau_c(V)b}. \qquad (8)$$

Table 2 Coefficient for runtime formula of single-broadcast[9]

	T3D		T3E	
Message transfer variant	$\tau(V)$	$T_c(V)$	$\tau(V)$	$T_c(V)$
PVM piecewise broad cast	53.65μs	0.0130μs	-18.99μs	0.0093μs
MPI broadcast	31.26μs	0.0153μs	96.96μs	0.0059μs
PVM accumulation with *max*	17.15μs	0.036μs	28.43μs	0.022μs
MPI accunlation with *max*	87.10μs	0.036μs	95.10μs	0.021μs

5. EXPERIMENTAL RESULTS

Here we mainly consider the partial differential equation taken from [1]
$Lu=f$, on $\Omega=(0,1)\times(0,1)$
with Dirichlet boundary condition u = 0 where

$$Lu = -\Delta u - 20\left(x\frac{\partial u}{\partial x} + y\frac{\partial u}{\partial y}\right),$$

and the right-hand side f is chosen so that the solution is

$$u(x,y) = \frac{1}{2}\sin(4\pi x)\sin(6\pi y).$$

Basically, we discretize the above differential equation using second order centered differences on a 400×400 with mesh size 1=441, leading to a system of 193600 linear equations with a nonsymmetric coefficient matrix of 966240 nonzero entries. Diagonal preconditioning is used. For our numerical tests, we choose x_0 = 0 as initial guess and tol = 10^{-5} as stopping parameter.
From the limited preliminary experimental results, the theoretical estimation gives very accurate prediction for the experimental results, the comparison shows that there is only 8% difference. The detailed experimental results is being carried out for the comparison and will be presented later.

6. REFERENCES

[1] H. M. B ucker and M. Sauren. A parallel version of the quasi-minimal residual method based on coupled two-term recurrences. In J. Wasniewski, J. Dongarra, K. Madsen, and D. Olesen, editors, Proceedings of Workshop on Ap- plied Parallel Computing in Industrial Problems and Opti- mization (Para96), LNCS184, Lecture Notes in Computer Science, pages 157{165. Technical University of Denmark, Lyngby, Denmark, Springer-Verlag, August 1996.

[2] H. M. B ucker and M. Sauren. A parallel version of the unsymmetric Lanczos algorithm and its application to QMR. Technical Report KFA-ZAM-IB-9605, Central Institute for Applied Mathematics, Research Centre Julich, Germany, March 1996.

[3] K. Cameron, L. J. Clarke, and A. G. Smith. CRI/EPCC MPI for CrayT3D. Technical report, Edinburgh Parallel Computing Centre, 1995.

[4] D. E. Culler, J. P. Singh, and A. Gupta. Parallel Computer Architecture: a Hardware Software Approach. Morgan Kaufmann, 1999.

[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] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. SIAM Journal on Scientific and Statistical Computing, 14:137{158, 1993.

[8] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. Numerische Mathematik, 60:315{339, 1991.

[9] T. Rauber and G. R unger. Modelling the runtime of scientific programs on parallel computers. In Workshop Proceedings of International Conference on Parallel Processing, Toronto, Canada, August 2000.

[10] Cray Research. Architectural overview of the CrayT3D, 1994.

[11] L. T. Yang and R. P. Brent. Quantitative performance analysis of the improved quasi-minimal residual method on massively distributed memory computers. Advances in Engineering Software, 33:169{177, 2002.

[12] 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), pages 170{177, March 1921, 1997. Shanghai, P.R.China.

[13] 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), pages 389{399, April 28-30, Vienna, Austria 1997.

[14] T. Yang and H. X. Lin. The improved quasi-minimal residual method on massively parallel distributed memory computers. IEICE Transactions on Information and Systems, E80-D(9):919{924, September 1997. Special issue on Architectures, Algorithms and Networks for Massively Parallel Computing.

[15] T. Yang and H. X. Lin. Isoefficiency analysis of CGLS algorithm for parallel least squares problems. In Proceedings of the International Conference on High Performance Computing and Networking (HPCN-97), pages 452{461, April 28-30, Vienna, Austria 1997.

[16] T. Yang and H. X. Lin. Solving sparse least squares problems with preconditioned CGLS method on massively distributed memory computers. Journal of Parallel Algorithms and Applications, 13(4):289{305, 1999.

A Parallel Iterative Solution of An Ill-Posed Problem in One-dimensional Heat equation with Source Term

Guo Qingping, Wang Weicang
Wuhan University of Technology, Wuhan 430063, P.R.China
qpguo@public.wh.hb.cn

ABSTRACT

Iterative method is popular for solving equations. There are many iterative methods that can be, and are, applied to ill-posed problems. Recently, we proposed a parallel iterative algorithm for solving an ill-posed problem in heat conduction, and we estimated the parallel efficiency of this iterative method.

Keywords: Parallel algorithm, Ill-posed problem, Discretization method, VBF

1. INVERSE HEAT CONDUCTION PROBLEM

In general, ill-posed problem in heat transient is called inverse heat conduction problem also. We pose the problem of finding a source term, $f(x,t)$, in the one-dimensional heat equation. That is, we deal with the partial differential equation

$$u_t = u_{xx} + f(x,t), \quad 0 < x < \pi, t > 0. \tag{1}$$

The term $f(x,t)$ represents a rate of production of thermal energy per unit time per unit length. For simplicity, we assume homogeneous boundary and initial conditions:

$$u(x,0) = 0, \quad 0 < x < \pi,$$

$$u(0,t) = u(\pi,t) = 0. \tag{2}$$

The problem we posed is the determination of the source term $f(x,t)$ from temperature measurements $u(a,t)$ at an interior site a, where $0 < a < \pi$.

To see the relationship between the source term and the interior temperatures, we will work formally with Fourier series. Suppose that $f(x,t)$ has a Fourier sine expansion

$$f(x,t) = \sum_{n=1}^{\infty} f_n(t) \sin nx,$$

Where the coefficients are given by

$$f_n(t) = \frac{2}{\pi}\int_0^{\pi} f(x,t)\sin nx \mathrm{d}x. \tag{3}$$

Assume a like expansion for $u(x,t)$,

$$u(x,t) = \sum_{n=1}^{\infty} u_n(t) \sin nx. \tag{4}$$

We find on substituting into (1) and using the initial condition in (2) that the coefficients $u_n(t)$ satisfying the nonhomogeneous linear initial value problems

$$u_n' + n^2 u_n = f \ ,$$
$$u_n(0) = 0, \tag{5}$$

A routine application of Laplace transforms solves (5) yielding

$$u_n(t) = \int_0^t \mathrm{e}^{-n^2(t-r)} f_n(r)\mathrm{d}r. \tag{6}$$

We note at this point that, for each given positive integer n, the coefficient $u_n(t)$ is purely temporal, however the entire collection of coefficients $\{u_n(t)\}$ contains all the spatial information in $u(x,t)$ as reflected in (4). From (6) we then see that the action of the exponential term will severely damp details in $f_n(t)$ and hence the recovery of the source term f from information in u will generally be a very difficult task.

To complete the analysis of the relationship between f and u, substitute (3) into (6) and use the result in (4). Interchanging the order summation and integration and substituting the interior point $x = a$ we find

$$u(a,t) = \int_0^t \int_0^{\pi} k(s,t-r) f(s,r)\mathrm{d}s\mathrm{d}r,$$

where

$$k(s,r) = \frac{2}{\pi}\sum_{n=1}^{\infty} \mathrm{e}^{-n^2\pi r} \sin na \sin ns.$$

We see an inverse problem phrased in terms of an integral equation of the first kind.

2. DISCRETIZATION METHODS

From previous section, we know that inverse heat conduction problem an integral equation of the first kind. There is general form for equations of this sort,

$$\int_a^b k(s,t)x(t)\mathrm{d}t = y(s). \tag{7}$$

We want to get numerical solution for inverse heat conduction problem. This requires that the problem be discretizaed, which is, expressed in terms of finitely many unknowns. The simplest way to accomplish this is to apply some quadrature rule, like the midpoint rule, Simpson's rule, etc, to the integral. Applying a quadrature rule with weights $\{w_j\}_{j=1}^n$ and nodes $\{t_j\}_{j=1}^n$ to (7) we obtain the approximate problem

$$\sum_{j=1}^{n} w_j k(s,t_j) x_j = y(s), \tag{8}$$

Where the number x_j are approximations to $x(t_j)$. Now (8) still represents an infinite system in that a constraint is specified for each of infinitely many values of s. Of course we can convert (8) into a finite dimensional problem by collocation, that is, by requiring (8) to holding at certain specified collocation points $\{s_i\}_{i=1}^m$,

$$\sum_{j=1}^{n} w_j k(s_i,t_j) x_j = y(s_i), i = 1,...,m. \tag{9}$$

In this way the integral equation (7) is approximated by the $m \times n$ linear system

$$Ax = b, \tag{10}$$

Where A is the $m \times n$ matrix with entries $w_j k(s_i,t_j)$, x is now an n-vector which is meant to approximate

$$[x(t_1),...,x(t_n)]^T$$

and

$$b=[y(s_1),...,y(s_n)]^T .$$

In discreted an ill-posed integral equation of the first kind an ill-conditioned linear system is produced. Generally, the finer the discretization, the closer the algebraic problem approximates the ill-posed continuous problem and hence the more ill-conditioned the algebraic problem becomes.

3. PARALLEL ALGORITHM

Using the multi-grid parallel algorithm to solve transient initial-boundary problem, Guo[1] solved the one-dimensional initial-boundary problem with VBF algorithm(VBF:Virtual Boundary Forecast). Furthermore, Guo[2] solve one dimensional ceramic/metal heat transient equation, the convergence speed was mach faster than that before. Wei[3] designed a parallel algorithm to get well speedup rate by suitable preconditioner reducing condition number of linear system.

Basing the algorithms listing before, we can design aparallel algorithm to iterative solve linear system (9) to approximate (7) as below:

1) Selecting weights, nodes and collocation points to get linear system (9) or (10).
2) Reordering the index of collocation points to reduce condition number of coefficient matrix of (10), that is, finding a invertible matrix P (by [3]) such that $cond(PA) < cond(A)$. And PA is blocked as form below

$$PA=\begin{matrix} A_{11} & A_{12} & \cdots & \cdots \\ A_{21} & A_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ A_{l1} & A_{l2} & \cdots & A_{ll} \end{matrix}.$$

3) Dividing the vector Pb, Px as l vectors $b_k, x_k (k=1,\cdots,l)$ following PA respectively.
4) Sending matrix A_{kk} and vector b_k (k=1,…,l) to processors respectively.
5) Adding uniform random errors ε_k to the right hand side with $|\varepsilon_k| < 10^{-4}$ and solving

$$A_{kk}x_k = b_k - \sum_{J\neq k} A_{kj}x_j, \quad k=1,...,l;$$

with multi-grid method by VBF([2]).
6) Denoting the global solution of (10) by x^*, and smoothing the solution x^* to get an approximation $x^{(1)}$.
7) Repeating the steps 2)—6) to get approximation $x^{(2)}$ from $x^{(1)}$.
8) Stopping at a suitable approximation $x^{(p)}$.

Convergence of this algorithm is obvious. Because VBF method ([2]) and parallel preconditioner algorithm([3]) are convergent, we can prove convergence of algorithm present as them.

4. PARALLEL EFFICIENCY

Suppose that node number on each processor is the same as others and it is n. If calculating amout of a V-cycle is denoted by wu, then working amount of multi-grid on a processor equals to

$$\frac{2}{1-2^{-1}} \cdot a \log(n) \cdot wu .$$

So the working amount of getting an approximation is

$$W_1 = \frac{2}{1-2^{-1}} \cdot a\log(nl) \cdot wu. \cdot$$

The working amount of multi-grid by serial algorithm is

$$W_2 = \frac{2l}{1-2^{-1}} \cdot a\log n \cdot wu. \cdot$$

Hence, the parallel efficiency of our algorithm is

$$E_l = \frac{W_1}{W_2} = \frac{\log(nl)}{l\ \log n} = \frac{1}{l} + \frac{\log(nl)}{l\ \log n} \cdot$$

5. CONCLUSION

From the discussing, we find it important to select suitable nodes and collocation points in difference ill-posed problems. And the discretization method adopted is simplest one. We want to find how to discrete difference inverse heat conduction problems([4]) and block coefficient, and to get robust parallel algorithm through numerical testing.

6. ACKNOWLEDGEMENT

This work was supported by Natural Science Foundation of China (No.60173046) and Science Foundation of Hubei Province (No.2000J153)

7. REFFRENCFS

[1] Guo Qingping et al, Parallel computing using domain decomposition for cyclical temperatures in ceramic/metal composites, London conference, 1998, London.
[2] Guo Qingping et al, Optimize algorithm of multi-grid parallel with virtual boundary forecast, Jour. Nume. and Comp. 2, 2000, Beijing.
[3] Wei Jianing et al, A multi-grid parallel algorithm of one dimensional virtual boundary ransack forecast, Jounal of Wuhan Transportation University, Vol.24,No.2,April 2000, Wuhan, China.
[4] Kirsch,A., Mathematical theory of Inverse Problem, 1997,Springer-Verlag, Berline.

An Access Control Architecture based on SPKI in Computing Grid

Li Bao-Hong Hou Yi-Bin Chen Xu-Hui
Department of Computer Science and Technology, Xi'an Jiaotong University
Xi'an, ShanXi, 710049, China
E-mail: bhli@citi.xjtu.edu.cn

ABSTRACT

Computing Grid is a large-scale distributed environment for collaboration. Because Grid is heterogeneous and dynamic, access control is a tough problem to be solved. In this paper, the requirement of access control in Grid is analyzed and an access control architecture based on SPKI is developed. In this architecture, global and local security mechanism can efficiently collaborate, and can implement Fine-grained access control. Meanwhile this architecture can help to reduce administrators and users' burden.

Keywords: Computing Grid, Access Control, SPKI, Authorization Certificate, SDSI Name Certificate.

1. INTRODUCTION

Computing Grid[1][2], which composes of supercomputers, storage systems, scientific instruments and other resources, is a large-scale distributed computing environment. It can couple individuals and institutions to form Virtual Organizations (VOs) in order to fully exploit network resources, enhance performance and scalability of systems. Computing Grid is distinguished from traditional distributed systems by the following characteristics. First, VOs include large shareable resources belonging to different administrative domains, each of which has its own security policy. Second, software and hardware in Grid are heterogeneous. Third, Computing Grid has the dynamic nature. For example, applications can dynamically request, use and release resources. Shareable resources and institutions can dynamically join and quit the VOs.

Characteristics of Computing Grid mentioned above lead to the complexity of security problem, especially the access control problem, because:

(1) The number of users and resources in Grids is enormous, and the security relationship is not simply between a client and a server, but among hundreds of entities locating at different administrative domains.

(2) Resources administrators usually use ACL to implement access control. Due to the dynamic of Grid, trust and collaboration relationships between different institutions keep changeful. In an institution, new users may join while some others may quit. So traditional methods of ACL place undue burden on administrators.

(3) The fact that applications in Grids can dynamically request, use, release resources makes it impossible to establish trust relationships among sites prior to applications execution, so complexity of access control is greatly increased.

(4) In Grids, different administrative domains may enforce different local security policies, such as Kerberos, DCE , SSH. So an access control architecture is essential to integrate diverse local security policies. For the purpose of security, this architecture can't replace local security policies, leaving local resources still managed by local administrators.

2. THE REQUIREMENT ON ACCESS CONTROL ARCHITECTURE

Based on the analysis above, it can be concluded that the access control architecture used by Grid should meet the requirements as listed below:

(1) Local security policy integrating frame. Administrative domains enforce different security policies, so the architecture should play a role of local security integrating frame. It can integrate diverse intra-domain access control technologies so that they can collaborate with each other. It should be emphasized that this access control architecture can't replace local security technologies, because local resources should still be administrated by local security policies.

(2) Scalability. When performance is considered, the overhead of administrating a VO (for example, adding or removing users, changing identities of users, etc.) shouldn't increase with number of users and resources in the VO. Because a institution can join several VOs，the overhead of administrating a resource increases only with the number of VOs it joins.

(3) Fine-grained access control. Although the number of users and resources in VOs is huge and may dynamically grow and shrink, the architecture can control every right of every user on the condition that the overhead of administration shouldn't notably increase.

(4) Security. The architecture ensures the security of Grids, especially it should prevent Grids from serious damage when credentials (certificates, passwords, etc.) are compromised.

3. SIMPLE PUBLIC KEY INFRASTRUCTURE

Simple public key infrastructure[3](SPKI for short), based on public key cryptography, is a proposed standard for access control. Compared with traditional public key infrastructure based on X.509, SPKI has below characteristics:

(1) Every user in SPKI can freely issue certificates to others. Traditional PKI's property that only Certificate Authorities (CAs) can issue certificates and CAs have hierarchical structure has some limitations: First, it is impractical that X.509 assumes that every user has a unique name (Distinguished name). On the contrary, SDSI[4] naming mechanism employed by SPKI can avoid this assumption; Second, users belong to different hierarchy should find common CA in higher level so that certificates signed by one lower–level CA can be trusted by others. This is also unpractical; Third, the cost of issue certificate revocation information is high, while SPKI can decrease this cost by reducing validity dates.

(2) By using SPKI authorization certificates, authorizations can be freely defined and distributed. For example, when Alice has the right to read, write and execute certain directory in file system, by issuing certificates she can give Bob the right to read and write this directory, and give Carl the right to only read this directory.

(3) In SPKI, rights can be transferred by delegation mechanism. For example, Alice can give Bob the right to read

and write certain directory, and Bob can also delegate this right to Carl.

(4) Key-oriented authorization certificates. To avoid the limitation of global unique name in X.509, SPKI uses key-oriented authorization certificates. This kind of certificate is shown below:

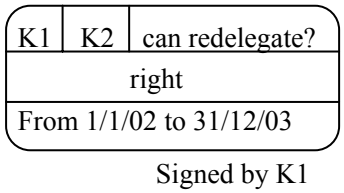


Figure1 Authorization certificate in SPKI

An authorization certificate contains signed 5-tuple: issue, subject, delegation, authorization, validity dates. Using this certificate, public key K1 authorizes certain right to public key K2 instead of Alice and Bob. Such key–oriented mechanism avoids the problem to find public key by users' distinguished name in X.509.

(5) SPKI uses SDSI name certificates to find public keys when necessary, without the limitation of global unique name. This mechanism is shown in figure 2.

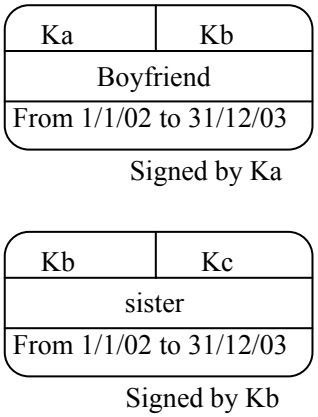


Figure2 SDSI's name certificate

For example, to find the public key of Alice's boyfriend's sister (Alice's public key is Ka), two SDSI name certificates in figure 2 are employed. The first certificate tells us the public key of Alice's boyfriend is Kb; The second certificate tells us the public key of Kb's sister is Kc.

(6) SPKI supports the concept of community certificates. Still take figure 2 for example. Perhaps Alice's boyfriend has several sisters, so these sisters have the same public key Kc and they own same rights authorized to Kc. Community certificates can greatly decrease administrating overhead of access control.

4. AN ACCESS CONTROL ARCHITECTURE BASED ON SPKI

Because SPKI's ideal characteristics mentioned above, an access control architecture for Computing Grid based on SPKI is developed in this paper. In this architecture, SPKI is used as global authentication mechanism.

The architecture is illustrated in figure 3. Institution 1 and 2 establish a VO for a collaborative planning. In every institution, several agents are set up for access control:

Local resource agent (LRA): Every shareable resource has a LRA. By using local security mechanism, it authenticates users to determine if they have proper access permission.

Global resources agent (GRA): A GRA in every institution is used to authenticate users by SPKI mechanism, the global security policy, and thus GRA s play a role to map global security policy to local ones.

Institution agent (IA): By issuing SPKI authorization certificates, IA administrates access rights of users within an institution. Users in an institution can be hierarchically organized, so a sub-institution agent (subIA) is used to administrate a group of users within an institution. In this case, IA issues authorization certificates to subIAs, and subIAs iteratively issue authorization certificates to users. IAs and subIAs can issue community certificates for multiple users to effectively reduce burden of user administration.

IA can greatly reduce the burden of GRAs. With the help of IA, GRAs can only administrate access rights of IAs within the VO, leaving the administration of individual users to IAs and subIAs. Meanwhile, SPKI mechanism can ensure that users can't acquire excessive rights, so security requirement can be met.

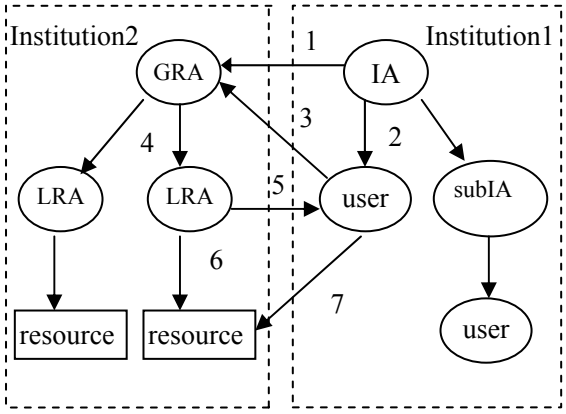


Figure 3 An access control architecture based on SPKI

Operations of this access control architecture are described below:

(1) During the establishment of VO, every IA, on behalf of every institution, registers in GRAs of other institutions, and acquire certain access rights. Meanwhile, LRAs allocate local credentials (such as userID, password) for IAs.

(2) By issuing SPKI certificates, IAs authorize rights to users. Some users may acquire total rights IAs own while others may acquire restricted rights, depending on roles they play.

(3) A user sends signed request to GRAs for allocation of resources. The request includes authorization certificates of user itself and IA in this institution. GRA can authenticate the user to determine if it has the proper permission.

(4) If the user has the permission, GRA, on behalf of the user, requests resources to LRA. LRA authenticates the user (in fact, the user'IA) using local authentication mechanism (for example, Kerberos, SSH and SSL). GRA's request should include necessary local credentials which are mapped by GRAs using mapping table.

(5) When local authentication succeeds, LRA grants access ticket to the user, and

(6) Allocate resources for the user.

(7) The user access resources using ticket acquired in step 6.

5. CONCLUSION

The architecture introduced above has obvious advantages: First, by using GRAs and LRAs, global and local access control mechanism can effectively collaborate; Second, IAs and subIAs can greatly reduce resources administrators' burden of access control; Third, SPKI community certificates can reduce the burden of user administration; Fourth, the requirement of fine-grained control can be completely met; Fifth, validity dates of SPKI certificate can be shorten to enhance security.

6. ACKNOWLEDGEMENTS

I would like to thank Prof. Hou and other colleague for many helpful discussions and for their extensive and insightful comments on various areas of this paper. I also thank the anonymous referees for providing valuable feedback.

7. REFERENCES

[1] Ian Froster, Carl Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgon Kaufmann.1999.

[2] Ian Froster, Carl Kesselman, Steven Tuecke. "The Anatomy of the Grid: Enabling ScalableVirtual Organizations". International Journal of High Performance Computing Application, 2001.15(3):pp.200-222.

[3] Carl M. Ellison, Bill Franz, Butler Lampson, Ron Rivest, Brian M. Thomas, Tatu Ylonen. "SPKI Requirements, SPKI Certificate Theory, Simple public key certificate, SPKI Example". Internet draft. October 1998.

[4] Ronald L. Rivest and Butler Lampson. "SDSI-A simple distributed security infrastructure". Technical report. April 1996.

Agent Based Grid Resource Management

Li Chunlin [1) 2)], Lu Zhengding[2)] , Li Layuan[1)]
Department of Computer Science, Wuhan University of Technology, Wuhan 430063,P.R.China[1)]
Department of Computer Science, Huazhong University Of Science &Technology, Wuhan 430074,P.R.China2)
E-mail: chunlin74@163.net or
jwtu@public.wh.hb.cn

ABSTRACT

This paper presents an Agent-based Computational Grid (ACG), which applies the concept of computational grid to agents. The ACG system is to implement a uniform higher-level management of the computing resources and services on the Grid, and provide users with a consistent and transparent interface for accessing such services. All entities in the Grid environment including computing resources and services can be represented as agents. Each entity is registered with a Grid Service Manager. An ACG Grid service can be a service agent that provides the actual grid service to the other grid member. The design and implementation of grid service discovery are given. Finally, related work and some conclusions are given.

Keywords: agent, service, grid, resource discovery

1. INFORMATION

Computational Grids are an enabling technology that permits the transparent coupling of geographically dispersed resources (machines, networks, data storage, visualization devices, and scientific instruments) for large-scale distributed applications. Grids provide several important benefits for users and applications: convenient interfaces to remote resources, resource coupling for resource-intensive distributed applications and remote collaboration, and resource sharing. Research in this burgeoning area embodies the confluence of high-performance parallel computing, distributed computing, and Internet computing, attracting successful research from all three disciplines. Several grid systems have been proposed in the last few years. Legion is an object-based project to develop software in support of a "World-wide Virtual Computer". The project envisions a system in which a user sits at a Legion workstation and has the illusion of a single very powerful computer. Globus is a large U.S. based project which is developing the fundamental technology that is needed to build computational grids such as execution environments that enable an application to integrate geographically-distributed instruments, displays, computational and information resources. Netsolve [8] is a client-agent-server paradigm based network enabled application server. It is designed to solve computational science problems in a distributed environment. The Netsolve system integrates network resources including hardware and software into the desktop application. It performs resource discovery and delivers solutions to the user in a directly usable and "conventional" manner (i.e., no need to develop special program code like parallel code to use high-end machines).
Agents can provide a useful abstraction at each of the three Grid layers. By their ability to adapt to the prevailing circumstances, agents will provide services that are very dynamic and robust, and suitable for a Grid environment. Agents can be used to extend existing computational infrastructures. Agent based Computational Grid (ACG) can offer a flexible service discovery and management.
Few research group has focused on offering an environment to combines the concept of computational grid and agents. In this paper, we propose an Agent-Based Computational Grid (ACG), which combines concept computational grid with agent. It provides a completely distributed environment within which agent systems and individual agents can participate in a broader community of agents, utilizing services and capabilities provided by other participants or the grid itself. ACG allows user/requestor agents to dynamically subscribe some grid services for a finite time period. A grid service can be subscribed before it can be configured or used by the user/requestor agent. The primary goal of the ACG system is to perform uniform higher-level management of the computing resources and services on the grid, and provide users with a consistent and transparent interface for accessing such services. The design goals of our model focus on providing a flexible and efficient grid resource discovery system to collaborate grid service agents in the grid.
The rest of the paper is organized as follows. Section 2 gives an overview of ACG grid. Section 3 describes the design of grid service management. Section 4 describes some related work. Section 5 concludes the paper.

2. THE OVERVIEW OF ACG GRID

The ACG Grid is intended to provide a completely distributed environment within which agent systems and individual agents can participate in a broader community of agents, utilizing services and capabilities provided by other participants or the Grid itself. ACG grid applies the grid concept to agents. It provides uniform access layer to a large variety of Grid services including other libraries and applications. In essence, ACG Grid can be viewed as a composition or federation of agent systems; it is responsible for providing services and allocating resources among its members and is used to make agent systems more interoperable.
The primary goal of the ACG system is to provide users with a consistent and transparent interface for accessing such services. All entities in the Grid environment including computing resources and services can be represented as Grid Service Agents. Each entity is registered with a Grid Service Manager. An ACG Grid service is a service agent that provides the actual Grid service to the user [20]. ACG grid provides a framework, which combines the capabilities of a computational grid and agent system architecture. As such, the grid would need to incorporate services provided by agent system architectures, such as communications, lifecycle services, matchmaking, facilitating, security, persistence facilities, system management, and mobility. ACG grid knows not only about agents, but also about their computational requirements

* The work is supported by National Natural Science Foundation of China and NSF of Hubei Province.

and about available computational resources. Hence, the ACG grid appears to incorporate both the concepts of the computational grid and the agent, it provides a unified, heterogeneous distributed computing environment in which computing resources are seamlessly linked. These agents play the roles of applications whose computations can be distributed within this distributed computing environment, resources that can be used within this environment, and infrastructure components of this environment. There is an interface between the computational and agent layers. Building the ACG grid involves all the computational grid issues of system management, distributed computation and load balancing, mobile code, security, etc. This requires a way of describing resources and capabilities, and resource requirements and tasks, and a way to map between them, at both agent and computational levels.

The ACG architecture can be viewed as comprising of a number of nodes capable of running grid service agents, service requestor agents and some infrastructure services. Grid members interact with each other to achieve a task. The service usage model is role based. To enable Grid Service Agents to be added, updated and removed easily from the network, some form of discovery system is required. This is the role of the "Grid Service Registry " which utilizes a database that can be searched to locate registered grid services. This Grid Service Registry provides a simple registration mechanism common to all nodes and a simple interface for searching the registered services common to all nodes; it also provides a mechanism for ensuring the registration database contains up to date information. It should be robust to guard against node failure, in meanwhile, it minimize the network traffic required for maintaining the registration system. Grid Service Registry utilizes a "Grid Service Manager" on every node in the grid. This manager is located on a fixed, pre-defined port, which is termed the "Registry Port" and provides a common access point into the Grid Service Registry for all nodes. By locating a server on all nodes, service requestor agents do not need any localized network information to access the Grid Service Registry. Every node has an individual Grid Service Manager, so the complete system is defined by the interaction between the service managers.

Grid Services are infrastructure services to facilitate ACG grid system. The core grid services are described as follows.

1) Grid Registration Service: It provides a mechanism to register services of various resources.

2) Grid Directory Service: it provides white pages service. The directory service is used to publish the location and attributes of all grid services registered with lookup services in the Grid.

3) Grid Discovery Service: The Discovery service is used by grid members to search for resources. The purpose of the Discovery Service is to select the "best" service, allowing the user to specify only the minimum parameters of interest. The Discovery Service first queries the Directory Service for a set of services that satisfy the client's request. The Discovery Service then filters the set of services according to the criteria in the request.

4) Access Service: The Access service is used to validate requests made by one grid member to another. Grid Service Manager receiving the request provides the requestor credentials, and then information about the request is passed to the Access Service to determine if the access is permitted. Not all actions need to be checked with the Access Service, only those actions which only some members are permitted to use.

3. THE IMPLEMENTATION OF GRID RESOURCE MANAGEMENT

The main actions involved in grid resource management are registry, discovery, subscription, and access. There are four Processes: Grid Service Registry Process (GSRP), Grid Service Discovery Process (GSDP), Grid Service Subscription Process (GSSP) and Grid Service Access Process (GSAP). GSRP is activated when Grid Service Agents register their presence by publishing their service descriptions. The GSDP provides a basic mechanism to discover grid services for requestor agents. GSSP is responsible for grid service detection and subscription. GSSP allows user/requestor agents to dynamically subscribe some services for a finite time period. GSAP is used for service requestor agent to access the located service.

3.1 Grid Service Registry Process

In order for a new grid service to become part of the ACG grid, the service must be known to a grid service manager. Service registration is the process to make the its service description available in the grid service manager. The Grid Service Registry Process (GSRP) is used to register grid services represented as Grid Service Agent (GSA) (See Fig.2). Each Grid Service Agent makes its services available for other grid member by registering the services to Grid Service Manager (GSM). Service registration is performed as follows. By sending out Grid Service Manager announcement over a range, a GSA can detect Grid Service Manager running within that range. Included in the Grid Service Manager announ- cement is an IP address and port number through which Grid Service Manager can contact the GSA. Grid Service Manager monitors the well-known port for service announcement packets of GSA. When Grid Service Manager receives a service announcement, it opens and inspects the packet. The packet contains information that enables the Grid Service Manager to determine whether or not it should contact GSA. If so, it contacts the GSA directly by making a TCP connection to the IP address and port number extracted from the packet. Using RMI, the Grid Service Manager sends an object, called a *Registry,* across the network to the GSA. The purpose of the *Registry* object is to facilitate further communication with Grid Service Manager. By invoking methods on this object, GSA can register its service as a part of service collection in Grid Service Manager. The GSA signs its service using *ServiceSigner*. The *ServiceSigner* stores a number of items, including the service public key, the service certificates, a computed code certificate, and a computed data certificate. The GSA sends its service to Grid Service Manager, now the service contains a bunch of security related information. The GSA calls *RegisterService()* or *UnregisterService()* to register or unregistered itself, respectively, with the local Grid Service Manager. Several important objects to implement service registry will be described in details.

Once GSA has a *Registry* object that is got from Grid Service Manager, GSA prepares to register its service. There are several classes that implement the GSRP: Servicebuilder, *ServiceSpace* interface and *ServiceHandler* class. The Servicebuilder registers grid services in the grid service manager. It is state-less. It just implements the method to register a grid service. Service-type specific Servicebuilder objects do the real work. When the Grid service manager started up, it instantiated those Servicebuilder objects and saved them in the ServiceRegistry object. When a *registerservice* () request is received, Servicetype attribute in the given service specification is used to determine the type of grid service. Based on the service type, the appropriate

Servicebuilder object is retrieved from the ServiceRegistry. Then the Servicebuiler invokes *CreateService()* method to create the new service reference. Before a service is registered, its ServiceDescription will be validated by the Servicebuiler. It is acceptable to have a null ServiceDescription, or a ServiceDescription with no ServiceAttributes. Both cases signify that this service cannot be discovered after registration. If the ServiceDescription contains one or more ServiceAttributes, the validation ensures that all attributes in a given Service-Description are consistent with the ServiceAttribute Descriptiom.

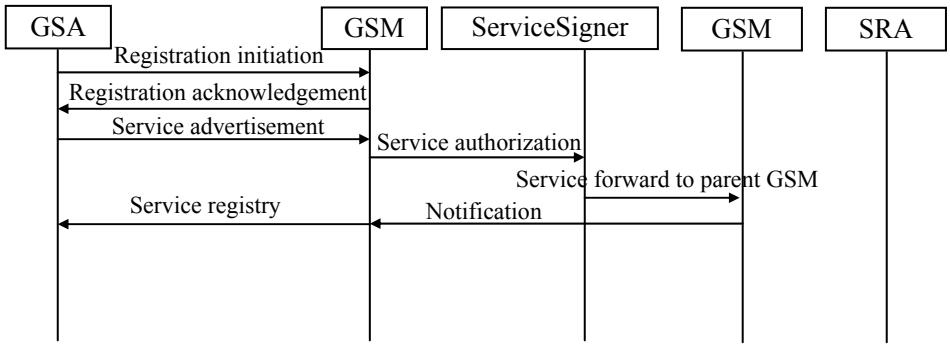


Fig.1 The process of GSRP

3.2 Grid Service Discovery Process

The Grid Service Discovery Process is used to discover any published resource or service. Resources or services are represented as advertisements. The Grid Service Discovery Process (GSDP) enables a service requester to find advertisements in its grid unit. The GSDP Process is the discovery Process of the Global Grid. Custom discovery services may choose to leverage GSDP. If a Grid Unit does not need to define its own discovery Process, it may use GSDP. The GSDP is implemented by three kinds of components: Grid Service Agents that provide services, Requestor Agents that request services and entities that constitute Grid Service Discovery Service itself.

Requests of Requestor Agent are expressed in terms of constraints. Constraints represent the requirements for the service to be found. In ACG, constraint language of service discovery is based upon XML. Constraints are passed as strings to the constructor of the XML class. During construction, the constraint is parsed and the expression tree is created. The Requestor Agent calls *SearchService()* to ask the local Grid Service Manager to search for Grid Service Managers containing registered services with a particular constraints. The local Grid Service Manager returns the list of GSM-IDs to the Requestor Agent. The Grid Service Manager with the GSM-ID included in the list has information about Grid Service Agent that can provide the service requested by the Requestor Agent. If Grid Service Manager is a parent manager it performs the search on the parent Grid Service Registry that should contain information on all registered services in the grid. If the local Grid Service Manager is a child manager, it will find a local service if a suitable one is available otherwise it will query a parent node for a suitable service. The searching process can also explicitly request a parent Grid Service Manager search. When multiple parent managers are available, the child manager has the option of selecting a parent based on some criteria such as node load.

3.3 Grid Service Subscription Process

The Grid Service Subscription Process (GSSP) is responsible for grid service detection and subscription (see Fig.3). GSSP allows user/requestor agents to dynamically subscribe some services for a finite time period. A grid service can be subscribed before it can be configured or used by the user/requestor agent. The Subscription Service object provides the functionality to locate and to subscribe/unsubscribe services. It contacts a Grid Service Manager in order to find available services, and interacts with the service subscription server. This interaction is necessary to prepare service usage in case of a grid service subscription. The service subscription server manages all related information of services subscribed by a user/requestor agent. This information is stored in a Service Subscription Object, which comprises a formal description of the service and user specific service data. After a grid service registers itself with a subscription server for the duration of time, a user agent can present a request to the subscription server to subscribe the service, the request specifies the parameters of the subscription: start time, duration, entities that participate in the subscription, and the enforcement mechanism used to control service access during the subscription. If the requested service is available, the subscription service then grants the subscription, notifying the "subscribed service" and the users on subscription start/stop. Service Subscription Server is responsible for the storage and retrieval of grid service specific data for subscribed services. By means of this server it is possible for the Grid Service Manager to determine, if the user agent has subscribed a certain grid service and to get hold of the corresponding user service subscription object.

For each subscribed grid service there exists a service subscription object, which comprises a reference to the grid service agent, from which the service was subscribed, the subscription duration (beginning/end of service subscription), a formal service description. If the subscribed grid service is a special type service, the Grid Service Manager contains an additional service description object, specifying the service. The grid service agent reference is stored directly in the service subscription object, which can be obtained via *getservice ()*. Via *listServices()* the Grid Service Manager can determine all services subscribed by the user agent. If an user agent requests a certain service, the Grid Service Manager searches for a suitable service by invoking *lookupService()*, which takes a service description object as argument. If one is subscribed, its related service subscription reference is returned, otherwise an exception occurs. The beginning and end of a service subscription is stored in a duration object. The start time can be obtained by invoking *getStartTime*, while the stop time may be determined by invoking *getStopTime()*.

SuscriptionServiceObject defines the interface between a subscribed service and a subscription service - a subscription service communicates with a subscribed service using a proxy

object which implements the methods of this interface. This is not a Remote interface, each implementation of a subscribed service exports proxy objects that implement the *Subscription-ServiceObject* interface local to a subscription service, using an implementation specific Process to communicate with the actual remote manager. All of the proxy methods obey normal RMI remote interface semantics except where explicitly noted. Two proxy objects are equal if they are proxies for the same subscription service.

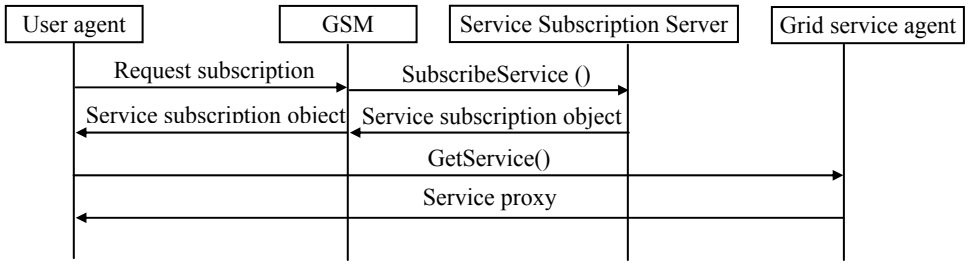


Fig.2 The process of GSSP

3.4 Grid Service Access Process

Grid Service Access Process (GSAP) is used for service requestor agent to access the located service (see Fig.4). Once a requestor agent receives the Reply Description of a Grid Service Manager, it can start communication with the GSM to use the services. Some methods and messages are defined to utilize a service provided by a Grid Service Manager. The local Grid Service Manager is the Grid Service Manager that the requestor agent is calling, and the remote Grid Service Manager is any other Grid Service Manager. The local Grid Service Manager may actually be in a remote node if the Requestor agent is calling the services through Remote Procedure Call.

When a Requestor agent wants to use a service provided by a Grid Service agent under a Grid Service Manager Process, it requests a Grid Service Manager to establish a Service Session**.** The Requestor Agent calls *ConnectService()* to ask the local Grid Service Manager to initiate a Service session with a specific Service agent registered at either the local Grid Service Manager or a remote Grid Service Manager. The Grid Service Manager, with which the specified grid services is registered, calls *ConnectService()* to notify the grid service agent of this *Connect Service* request. The grid service agent may either accept or reject the request. The result is communicated back to the Requestor Agent through the return parameter of the *ConnectService()* call. Once the Service session is established, the Requestor Agent calls *SendMessage()* to ask its respective local Grid Service Manager to send Requestor Agent-specific data to the other end of the Service session. The Grid Service Manager at the other end calls *ReceiveMessage()* to pass the Requestor Agent-specific data received from the other end of the Service session. After Connection session completes, the Requestor Agent calls *EndConnection()* to ask the local Grid Service Manager to terminate the Service session. The specified Grid Service Manager calls *EndConnection()* to notify that the Service Session is terminated.

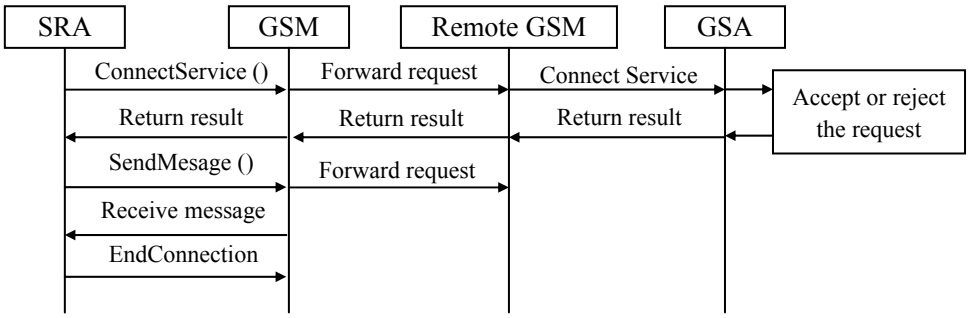


Fig.3 Grid Service Access process

4. RELATED WORK AND DISCUSSION

There several important computational grid projects that inspire ACG grid.

Globus[14] provides software infrastructure that enables applications to view distributed heterogeneous computing resources as a single virtual machine. Globus is constructed as a layered architecture in which higher-level services can be developed using the lower level core services. Globus offers Grid information services via an LDAP-based network directory called Metacomputing Directory Services (MDS).

The DARPA CoABS (Control of Agent Based Systems) project [15] aims to develop agent systems for offering specialized services, such as component interaction managers, database wrappers, traders/brokers, to resource planners and interactions managers. The CoABS Grid provides a framework for integrating diverse agent-based systems in a robust and agile manner. Agents and object-based systems dynamically register and discover relevant participants in order to collaborate meaningfully.

A4 (Agile Architecture and Autonomous Agents) methodology [16], which can be used for resource management for grid computing. An initial system implementation utilizes the performance prediction techniques of the PACE toolkit to provide quantitative data regarding the performance of complex applications running on local grid resources. At the

meta-level, a hierarchy of identical agents is used to provide an abstraction of the system architecture. Each agent is able to cooperate with other agents to provide service advertisement and discovery to schedule applications that need to utilize grid resources. A performance monitor and advisor are in development to optimize the performance of agent behaviors.

Condor [17] is a high-throughput computing environment developed at the University of Wisconsin at Madison. It can manage a large collection of computers such as PCs, workstations, and clusters that are owned by different individuals. The Condor *collector*, which provides the resource information store, listens for advertisements of resource availability. A Condor resource agent runs on each machine periodically advertising its services to the collector. Customer agents advertise their requests for resources to the collector. The Condor matchmaker queries the collector for resource discovery that it uses to determine compatible resource requests and offers.

Legion [18] is an object-based metasystem or Grid operating system developed at the University of Virginia. Legion provides the software infrastructure so that a system of heterogeneous, geographically distributed, high performance machines can seamlessly interact. Legion provides application users with a single, coherent, virtual machine.

Compared with above grid projects, our ACG grid provides a uniform higher-level management of the computing resources and services on the Grid, and provides users with a consistent and transparent interface for accessing such services. All entities in the Grid environment including computing resources and services can be represented as agents. An ACG Grid service can be a service agent that provides the actual grid service to the other grid member. Grid members communicate with each other by communication space that is an implementation of tuple space. ACG supports grid service subscription that allows user/requestor agents to dynamically subscribe some grid services for a finite time period.

5. CONCLUSIONS

In this paper, we have presented an agent based computational grid. This paper mainly describes grid service discovery architecture and basic design, how service requestor agents locate specific grid service agent by submitting requests to the Grid Service Manager with descriptions of required services in the network, and how Grid Service Agents dynamically register their services in Grid Service Manager. Some basic performance evaluations are made. The model has been implemented in a prototype, some practical application work based on it remains to be done in the future.

6. ACKNOWLEDGEMENTS

The work is supported by National Natural Science Foundation of China and NSF of Hubei Province**.**

7. REFERENCES

[1] I. Foster and C. Kesselman, The Grid : Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999.

[2] F. Kon, T. Yamane, et al., Dynamic Resource Management and Automatic Configuration of Distributed Component System, 6 th Usenix Conference on Object-Oriented Technologies and Systems(COOTS'2001)February 2001.

[3] Li Chunlin, Li Layuan, An Agent-oriented and Service-oriented Environment for Deploying Dynamic Distributed Systems, Journal Computer Standard and Interface, Elsevier, Vol 24/4 pp. 38-51, Sept, 2002.

[4] Li Chunlin, Lu Zhengding, Li Layuan, Coordinating Mobile Agents By XML-Based Tuple Space, Journal of Computer Science and Technology, Vol 17, No.6, 2002.

[5] Nick Antonopoulos, Alex Shafarenko. An Active Organization System for Customized, Secure Agent Discovery, The Journal of Supercomputing, 20, 5–35, 2001.

[6] Li Chunlin, Lu zhengding, Li layuan. Design and Implementation of a Distributed Computing Environment Model for Object_Oriented Networks Programming, Journal of Computer Communications, Elsevier, Vol 25/5, pp 517-522, Mar 2002.

[7] R. Buyya, S. Chapin, D. DiNucci, Architectural Models for Resource Management in the Grid, First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Springer Verlag LNCS Series, Germany, Dec. 17, 2000.

[8] R. Buyya, D. Abramson, J. Giddy, Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid, International Conference on High Performance Computing in Asia- Pacific Region (HPC Asia 2000), Beijing, China. IEEE Computer Society Press, USA, 2000.

[9] Li Chunlin, Lu Zhengding, Li Layuan , Zhang Shuzhi, A Mobile Agent Platform Based On Tuple Space Coordination, Journal of advances in engineering software, Elsevier, Vol 33/4, pp. 215-225, April 2002.

[10] [10] Li Chunlin, Lu zhengding, Li layuan. A Distributed Computing Model And Its Application, IEEE ICCNMC 2001, IEEE Computer Society Press, pp. 341-346, Oct, 2001.

[11] O. F. Rana, and D. W. Walker, The Agent Grid: Agent-Based Resource Integration in PSEs, in Proc. of 16 th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation, Lausanne, Switzerland, 2000.

[12] K. Krauter, R. Buyya, and M. Maheswaran, A Taxonomy and Survey of Grid Resource Management Systems, Software: Practice and Experience, 2001.

[13] O. F. Rana and Luc Moreau, Issues in Building Agent-Based Computational Grids, UK Multi-Agent Systems Workshop, Oxford, December 2000.

[14] I. Foster and C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit., Intl Journal of Supercomputer Applications, Volume 11, No. 2, 1997.

[15] The CoABS DARPA Project, Control of Agent Based Systems, http://coabs.globalinfotek.com/.

[16] J. Cao, D.J. Kerbyson, G.R. Nudd. Performance Evaluation of an Agent-Based Resource Management Infrastructure for Grid Computing, Proceedings of 1st IEEE/ACM International Symposium on Cluster Computing and the Grid. Brisbane, Australia, 311-318, May 2001.

[17] J. Basney and M. Livny, Deploying a High Throughput Computing Cluster, High Performance Cluster Computing, Vol. 1, Chapter 5, Prentice Hall PTR, May 1999.

[18] S. Chapin, J. Karpovich, A. Grimshaw, The Legion Resource Management System, Proceedings of the 5 th Workshop on Job Scheduling Strategies for Parallel Processing, April 1999.

Ubiquitous Aware Computing for Building Smart Home in Ambient Intelligence

Xuhui Chen Yibin Hou Xue Qin Baohong Li
Computer and Information Technology Institute, Xi'an Jiaotong University,
Xi'an , Shanxi, 710049, People's Republic of China
E-mail: xuhui.chen@163.com

ABSTRACT

This paper presents the design and analysis of a conceptual framework with specific knowledge database based software Agent for supporting context aware home net. The context aware computing implements all of resources connected in the residential home in Ambient Intelligence, including storage, computing, communication and knowledge resources. Ultimately, sharing resource and cooperating with job in the virtual home network. Users are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects and an environment that is capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way. The design principle of the aware home network is simple and smart: it only adds two major software entities to a seamless mobile/fixed communication infrastructure: smart agent and Context Active Knowledge Base (CAKB) which contains user's physical, social, and psychological environment information. The CAKB can be distributed among different servers, User's environment data is sensed by the smart multimedia terminals and stored in the CAKB. Up on a request, the smart agent searches in the CAKB, analyzes and presents its user's context information to the request. The multimedia terminal is smart card driven. Functions of the multimedia terminal including eyeglass-base display, headphone, wearable keyboard and radio devices are distributed around body where they are most convenient for the user and the devices communicate with each other through a HNAmI (Home Net in Ambient Intelligence).

Keywords: Ubiquitous Computing, Context Aware, Aware Home, Ambient Intelligence, Active Database..

1. INTRODUCTION

At present, there is a heterogeneous and distributed nature of home network, it contains data network, the video A/V network and control network. Because of the existence of various control standard and specification, control network also presents heterogeneous tendency. Therefore, system integration can implement the infrastructure of the home net.
The infrastructure of system integration has some problems. There is not uni- form specification, consumer may add any mixture and variety of devices to home net, a developer must have some means of tracking and interacting with the devices on the home work, so system integration is very complex for the variety of protocols and medium. Obviously, the data information and resource are isolated in the home1,2.

With the advances of wireless communication, the future communication must integrate Service and Network, and the consumers could easy connect to the Internet who- ever, whenever, wherever and whatever they want to do. One challenge of mobile distributed to exploit the changing environment with a new class of applications that are ware of the context in which they are run. The Context Aware Home is a residential laboratory where the future of homes is being developed and tested in our research. In the home there exists many sensors that record and analyze data. The result is a home that is more helpful or aware. The many tools being developed for homes could be some of the most helpful additions to many people lives. Also, these additions to homes could greatly aid the advancement for medicine by gathering vast amounts of long-term data about lifestyle that can be correlated with information about disease. Unfortunately there is a large downside to the existence of such sensors in a home, privacy. Every single piece of information gathered in a person's private residence is likely to be considered private. So, with the capability for any or all of this to be recorded by different means there exists an issue of privacy. For every sensor there is a different debate about where it should be what it should do and the payoffs of having the device.
Our goal in this paper is to propose an infrastructure, which is flexible enough to integrate a variety of different home nets with the rapidly developing context aware computing. The context aware computing implements all of resources connected in the residential home in Ambient Intelligence, including storage, computing, communication and knowledge resources. The design principle of the aware home network is simple and smart: it only adds two major software entities to a seamless mobile/fixed communication infrastructure: smart agent and Context Active Knowledge Base (CAKB), which contains user's physical, social, and psychological environment information.

2. SCENARIOS — MARIA'S AWARE HOME

In the Maria's Home, they are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects and an environment that is capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way. There currently exists an array of cameras in the ceiling, which are used for tracking the position of bodies in rooms. This positioning system is an automated one where camera data is sent to a computer for processing.
Microphones are also currently in place in the Aware Home for the purpose of an intercom system. For a house where two elderly people live they my find it rather convenient to be able to just speak in any room and be heard by the other person they want to speak to.
Pressure sensors are components that can be placed in the feet of a chair of the seats of a couch. They could help a house system detect where people are and what they are doing without giving away identity.
An Infrared Vector Tracking System is under development where infrared or visible light beams are shot and received throughout the house. This is done with to close and parallel beams that when crossed can give a computer the necessary

information to track peoples movement through a house.
The Aware Floor is a floor made up of panels that senses footsteps in an area. As a person walks they set down and lift up their feet in a particular and unique way. This creates a foot signature. Thus, an aware floor cannot only tell you where a person is standing or walking but it can identify them at the same time.
Installed in all the kitchen cabinets and the refrigerator are micro switches that detect whether the cabinets are open or closed. These were installed for the purpose of developing a system to help people who cook to resume where they stopped if interrupted.
Biometrics identification methods can be used to grant access to the house, but also have some issues in privacy when the access events are time stamped. If each event is stamped with a time, then the system will know how long a person is in the bathroom on average, or what time they generally leave the house.
Similar to corporate environments is the issues surrounding computer access in the aware home. By looking at large companies one can see that there are no doubt many automated methods for monitoring of computer usage. This includes web browsing, email, newsgroups and all other communications over the Internet in addition to activities that are performed solely on the local terminal such as word processing and such.
The Radio Frequency Identification system (RFID) in the aware home allows the detailed tracking and recording of any tagged object or entity throughout the house.

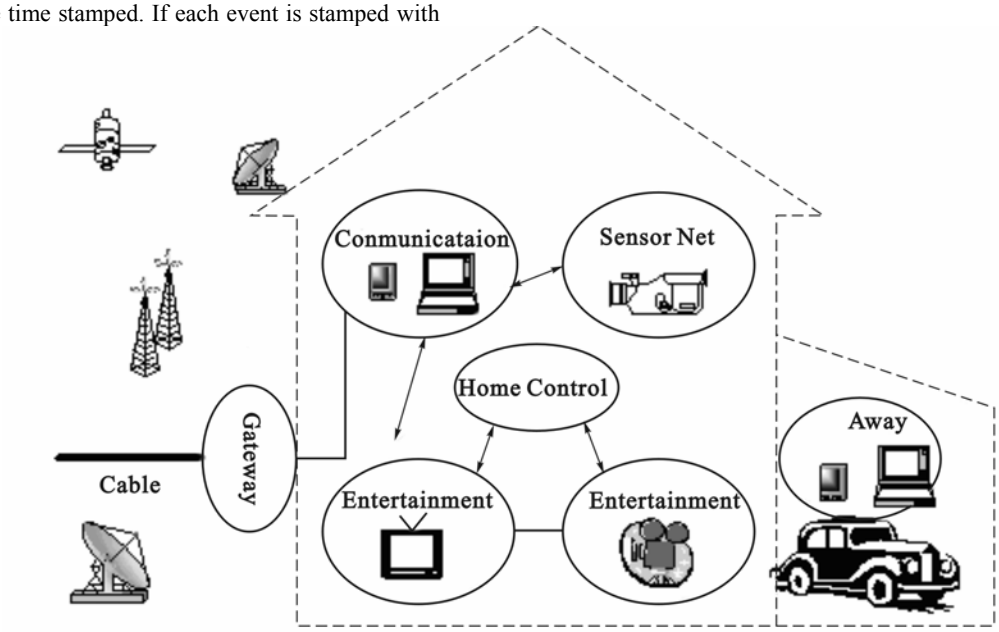


Figure 1 The Aware Home

3. UBIQUITOUS/CONTEXT AWARE COMPUTING

In our aware home there exists many sensors that record and analyze data. The result is a home that is more helpful or aware. Such context-aware systems adapts according to the location of the use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. A system with these capabilities can examine the computing environment and react to changes to the environment. Three important aspects of context are where you are, whom you are with, and what resources are nearby.
The terms “Ubiquitous computing, as described by Mark Weiser is a system where computing fades into the background and you don't notice it anymore. Context awareness is a theme within ubiquitous computing. Context-awareness is a feature of applications where systems can adapt to the user's context.” Anind Kumar clarified this things.
Context aware environment aware, and location aware computing can be found in different literature with different definitions. Abowd defines context aware computing as work that leads to the automation of a software system based on knowledge of a user's physical, social, emotional, or informational state. We believe that a user's context should basically include all of this information and the relation of this information about the user's physical environment, such as social relations, social position, and group relations. As well as the user's psychological situation, such as, emotional state, happiness/unhappiness, if they are busy, etc.3-9
The context-aware computing cycle, consisting of discovery (dynamic data), selection (location-based information), and use (application adaptation), is described next. Following this is an exposition on the types of information for use by mobile applications along with how that information can be used.
Most of current work on context-aware computing is concentrated on the physical environment awareness. But perhaps the most successful application of context aware computing in the future will be systems that could be use knowledge of a user's social and psychological environment information in combination with user's physical environmental information to support their communication and control various devices. We use a framework with an addition of a context information base and an active knowledge base for reasoning.

4. A CONCEPTUAL FRAMWORK FOR SUPPORTING AWARE HOME.

Aware home net is a network architecture in which the network nodes have the capability to sense, to reason and to be aware of the context and behavior of users. It automatically provides active services to users according network differs form an active service network in the way that it can sense and reason. It also differs from an active network in the sense that it separates the network transport function and service

function into different logical layer. Figure 2 show a smart network framework, which consists of different logical layer: network, service, knowledge, proxy, and smart terminal1.
The network layers consist of interconnecting networks including backbone networks and wired or wireless access network. This so-called "underlying network" provides basic capabilities by means of links, routers, and switches – to transport data packets from end to end.

The active service layer provides application services. The services can be recon- figured for different users and even for the same user at different times or for different situations. It may include different servers, such as badge serves, www servers, information service, data base services, etc. which provide application services. The sensor service of the network belongs to the service layer.
The Active Context Knowledge Base (ACKB) layer is built upon the service layer. It is a logical layer, which contains the context knowledge of the networks and users. It may include different knowledge base, reasoning algorithms, intelligence agent, etc.
The Middleware layer contains three logical components: device agents, active maps, and user agents. Each of these components is based on the dynamic environment publish subscribe communications model the devices agent and user agent. Device agents manage device-objects that are published in the active map or the appropriate user agent if they are "owned". One important attribute contained in the device object is location. The active map allows searches over located-objects registered within a particular location. Locations are defined as part of a containment hierarchy and are therefore nested. The active map knows about a number of spatial relations: containment, exact location, co-location, path distance and paths. Active maps are a meeting place where clients can find each other as well as relevant located objects. The design of active map service relies on a clustering of location information into a number of servers. The third component in the architecture is the user agents that encapsulate user specific information. A user agent exports a user environment and publishes a user-object. The user environment contains all of the devices in use as well as user-global parameters and customizations. The user agent design incorporates a back end server and a number of front-end fragments run on each host executing a user's application.

Smart terminals are the end system components that can communicate the smart homework and access the network services. A smart terminal should have sensors and provides context information concerning it users, user profiles, location and other related information to the network.

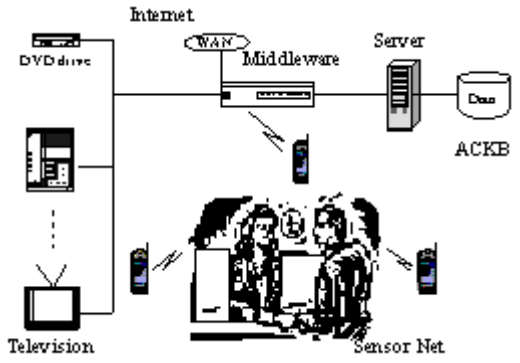


Figure 2 Architecture of Aware Home

5. CONTEXT AWARE HOME APPLICATION DESCRIPTION

In this chapter, we will discuss a number of different applications that have been designed and implemented with the Context Aware Computing. Our goal is to demonstrate that our architecture can support applications. The applications we will present are shown in Figure 3. It contains Sensor Server, Identity Broad, Intercoms, Recall Server, and so on.
Maria, Tom and their child Bob are living in the Aware Home in Ambient intelligence. It is six o'clock in the afternoon; Maria and Tom are taking a coffer at their living room. They don't want to be excessively bothered during this pause. Nevertheless, all the time they are receiving and dealing with incoming calls and mails.
At 6:10 p.m. A sales woman comes the door and wants sales some new production to them. To broadcast a message to all other occupants, she can say, "House, I would like to introduce …". An array of cameras and microphones sensor in the door should collect this audios and video information to the middleware. And the middleware processes the available data from ACKB, reasoning that this production cannot suit for them, so it can offer information to the sensor. The audio announcement is delivered to the saleswoman "Sorry, the production cannot suit for me ".
And now, Bob is playing in the yard, a five-year-old boy. Maria is worried about him, she wants to use the intercom to facilitate baby monitoring. Intercoms in homes are intended to facilitate conversations between occupants distributed throughout the home. She can say, "House, how is the baby doing? " The intercom delivers the audio from the room to the yard. "Fine, Mum " said the Bob.

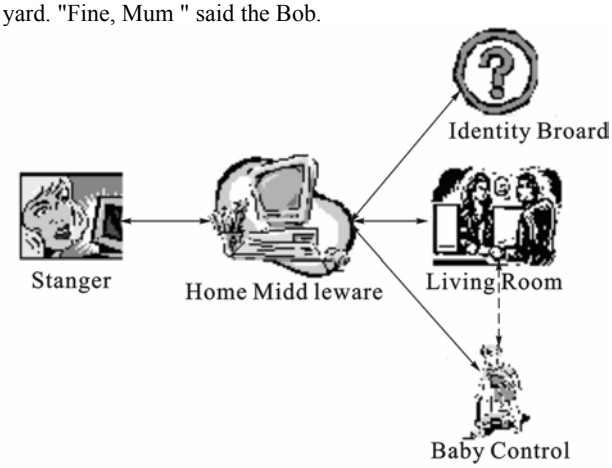


Figure 3 Context Aware Home Application. It contains Sensor Server, Identity Broad, Intercoms, recall Server, and so on.
Together these applications cover a large portion of the design space of context-aware applications, using time, identity, location, and activity as context and perform the context-aware features of presenting context information, automatically executing a service based on context, and tagging captured information with context to promote easier retrieval.

6. IMPLEMENTATION OF THE CONTEXT AWARE APPLICATION

The application features in the above scenario presented have all been implemented. The designs of the implementation applications we will present are shown in Figure 4. We use a framework with an addition of a context information base and

an active knowledge base for reasoning. The design principle of the aware home network is simple and smart: it only adds two major software entities to a seamless mobile/fixed communication infrastructure: smart agent and Context Active Knowledge Base (CAKB), which contains user's physical, social, and psychological environment information.10

The Context Active Knowledge Base contains both a context information base and an active knowledge base.

The Context Information Base

The context information base is standard database containing a user's physical and psychological environment information. The context information base can be distributed among different servers, User's environment, context data, in this case their location data, sensed by the smart devices and stored in the context information base. Up request, the agent searches the context information base, analyzes and presents its user's context information in response to the request.

The database can only provide information about the users physical and psychologycal environment information, such as location, position, happiness, etc. It does not draw any relation between the information. The most important use of this information is to support reasoning about the behavior of users based on this information. For example, if a user is in a conference room according to the location information. Then she/he is most likely in a meeting, then the incoming call should be forwarded to voice mail or the caller could be told to send a fax or email. Reasoning about these relations and structuring the information is the responsibility of the knowledge base.

Active Knowledge Base:

The knowledge abstracts relation based on the context information base. It drives the relations concerning this information. An active Knowledge Database contains Database System, Event Base and Event Monitor.

It forms as: $AKDB = DB + EB + EM$

Then reasoning algorithm is very simple ECA-rules. ECA-rules (event-condition- action rules) consist of events, conditions and actions. The meaning of such a rule is: "when an event occurs, check the condition and if it holds, execute the action".

Active ECA-rules description:

Rule name [<parameter >]

Events

When <condition>

IF [E_C_mod] Then Actions

IF [C_A_mod] Then Actions

Scheduling method

FAILURE Action

FAILURE transaction.

End Rule [<>]

Once a set of rules has been defined, the active database system monitors the relevant events. Whenever it detects the occurrence of a relevant event it notifies the component responsible for rule execution of it. We call this notification the signaling of the event. Subsequently, all rules that are defined to respond to this event are triggered (or fired) and must be executed. Rule execution incorporates condition evaluation and action execution.

Event Monitor estimates the information according with the user events or system events, which the information comes from the Active Service Component. If it accord, trigger the corresponding event and put in the event queue. Event Monitor can record the system event and information; it also maintains and checks the system log. Time Trigger sends out the system clock to event monitor.

Condition Evaluation forms Event-Condition- Coupling Mode (E-C-mod) or Condition-Event- Coupling Mode (C-E-mod) based on Events and corresponding Rules, and combines E-C-mod or C-E-mod as the conditions. If an event occurs, check the condition table and if it holds, execute the action based on priority scheduling method.

Action Trigger. When an event occurs, execute the correlation action; Action Trigger schedules the Action queue base on the action name and parameter. If the executing action is failure, Action Trigger can cancel the action.

Event and Rule Manager maintenance and manages the events and rules that has been defined and stored in the database. Event base and rule base can be recalled in any moment and alter.

7. CONCLUTION AND THE FUTURE WORK

With the advances of wireless communication, one challenge of mobile distributed to exploit the changing environment with a new class of applications that are ware of the context in which they are run. We present the design and analysis of a conceptual infrastructure with specific knowledge database based software Agent for supporting context aware home net. The context aware computing implements all of resources connected in the residential home in Ambient Intelligence, including storage, computing, communication and knowledge resources. Ultimately, our intention is sharing resource and cooperating with job in the virtual home network. Users are surrounded by intelligent intuitive interfaces that are embedded in all kinds of objects and an environment that is capable of recognizing and responding to the presence of different individuals in a seamless, unobtrusive and often invisible way.

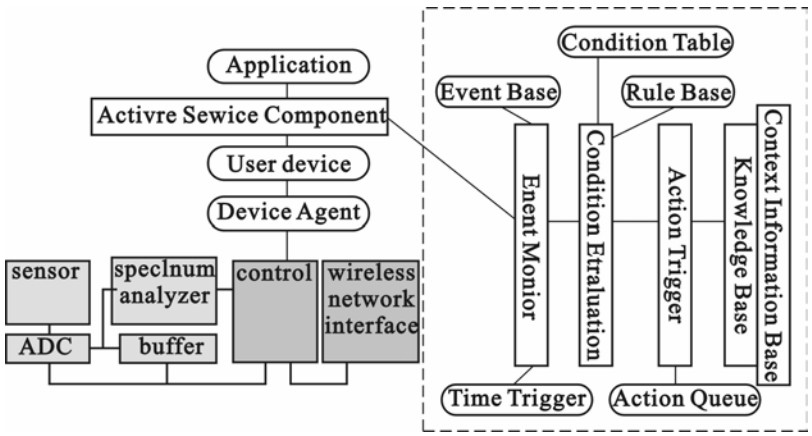


Figure 4 Implementation of the Aware Home Applocation

Our design principle of the aware home network is simple and smart: it only adds two major software entities to a seamless

mobile/fixed communication infrastructure: smart agent and Context Active Knowledge Base (CAKB), which contains user's physical, social, and psychological environment information. The CAKB can be distributed among different servers; User's environment data is sensed by the smart multimedia terminals and stored in the CAKB. Up on a request, the smart agent searches in the CAKB, analyzes and presents its user's context information to the request. The multimedia terminal is smart card driven. Functions of the multimedia terminal including eyeglass base display, headphone, wearable keyboard and radio devices are distributed around body where they are most convenient for the user and the devices communicate with each other through a HNAmI (Home Net in Ambient Intelligence).

8. REFERENCE

[1] A.K. Dey, D. Salber, and G.D. Abowd. Context based Infrastructure for Smart Environments, Proc. 1st Int'l Workshop on Managing Interactions in Smart Environments (MANSE 99), Springer-Verlag, New York, 1999, pp. 114–128.

[2] A.K. Dey et al. An Architecture to Support Context-Aware Applications, tech. report GIT-GVU-99-23, Graphics, Visualization, and Usability Center, Georgia Tech, Atlanta, 1999.

[3] T. Starner, B. Schiele, and A. Pentland. Visual Contextual Awareness in Wearable Computing, Proc. 2nd Int'l Symp. Wearable Computers (ISWC 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 50–57.

[4] K. Van Laerhoven and O. Cakmakci, What Shall We Teach Our Pants?, Proc. 4th Int'l Symp. Wearable Computers (ISWC 00), IEEE CS Press, Los Alamitos, Calif., 2000, pp.77–83.

[5] G. Kortuem, Z. Segall, and M. Bauer. Context-Aware, Adaptive Wearable Computers as Remote Interfaces to 'Intelligent' Environments, Proc. 2nd Int'l Symp. Wearable Computers (ISWC 98), IEEE CS Press, Los Alamitos, Calif., 1998, pp. 58–65.

[6] Anind K. Dey. The What, Who, Where, When, and How of Context Awareness, report College of Computing, Georgia Institute of Technology, Atlanta, Georgia.

[7] http://www.tangis.com. Context Aware Computing, A Tangis white paper on the future of mobile/wireless computing,

[8] Mandato D, Kovacs E. CAMP: a context-aware mobile portal , IEEE Comm- unications Magazine , Volume: 40 Issue: 1, Jan. 2002 pp 90 -97

[9] Anhalt J, Smailagic A. Toward context aware com- puting: experiences and lessons, IEEE Intelligent Systems [see also IEEE Expert] , Volume: 16 Issue: 3 , May-June 2001 pp 38 -46

[10] Schmidt A.,van Laerhoven K. How to build smart appliances?, IEEE Personal Communications , Volume: 8 Issue: 4 , Aug. 2001,pp 66 -71

[11] Joong-Han Kim, Sung-Su Yae. Context-aware application framework based on open service gateway, Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on , Volume: 3 , 2001pp 209 -213

[12] Yau, S.S.; Karim, F. Context-sensitive middleware for real-time software in ubiquitous computing environments, Object Oriented Real-Time Distributed Computing, 2001. ISORC - 2001. Proceedings. Fourth IEEE International Symposium on , 2001,pp 163 -170

[13] Kovacs E, Rohrle K,.Schiemann B. Adaptive mobile access to context aware services, Agent Systems and Applications, 1999 and Third International Symposium on Mobile Agents. Proceedings. First International Symposium on , 1999 pp 190 –201.

An Efficient Mapping Strategy for Task Scheduling on Multiprocessors

Jun Sun, Lijuan Zhu, Wenbo Xu, Ling Bao
School of Information Technology, Southern Yangtze University
Wuxi, Jiangsu Province 214036, China PR
E-mail: sunjun21c@163.com

ABSTRACT

In existed heuristic algorithms, list scheduling is an important class of mapping strategy for task scheduling on multiprocessors. It can produce more satisfactory solutions at significantly lower cost than other approaches. But this approach is only used for a bounded number of processors. As for the processors with unlimited number, the mapping strategy is based on cluster scheduling. In this paper, we bring the list scheduling method into the tasking allocation problem on an unbounded number of processors. The proposed algorithm is called Node-Transferring Scheduling Algorithm (NTSA), of which the experiment results show it outperforms other algorithms of the same problem in solution quality.

Keyword: task graph, scheduling, multiprocessor, critical path, node-transferring.

1. INTRODUCTION

In static scheduling problem, a parallel program to be executed on a multiprocessor system is modeled by a directed acyclic graph (DAG), in which nodes represent the tasks and edges represent the communication cost as well as the dependencies among the tasks. The objective of an efficient scheduling algorithm is to minimize the schedule length of the parallel program. As scheduling problem have been proven to be NP-complete [4], heuristics is devised for obtaining suboptimal solutions in affordable amount of computation time. An important class of scheduling algorithms is list scheduling which used for a bounded number of processors. (e. g MCP [5], DPS [8]). Generally, there are two steps as a list-scheduling algorithm is performed.

(1) Acquire a scheduling list by some priority strategy.

(2) Remove every node and schedule them to the target processors by the order of the scheduling list. This step is also addressed as mapping the task graph to the multiprocessors.

As of scheduling problem for an unbounded number of processors, algorithms usually schedule a DAG to an unbounded number of clusters. The clusters generated by these algorithms may be mapped onto the processors using a separate mapping algorithm. (e. g., LC[10], DCP[7], DSC[6]). In this paper, we propose a list-scheduling algorithm for an unbounded number of processors. The algorithm has good performance and tolerable time complexity as well.

The remaining paper is organized as follows: Section 2 defines the scheduling problem and introduces some definitions used in the paper. Section 3 describes our proposed algorithm. Section 4 includes a scheduling example illustrating the operation of the algorithm. Next, we present the experiment results in Section 5, and conclude the paper with some remarks in Section 6.

2. PRELIMINARIES

A parallel program can be modeled by a directed acyclic graph (DAG) $G=(V, E)$, where V is a set of v nodes, representing the tasks, and E is a set of e directed edges, representing the communication message. Edges in a DAG are directed and, thus, determine the precedence constraints among the tasks. The cost of node n_i, denoted as $w(n_i)$, represents the communication cost of the message. The source node of an edge (n_i, n_j), denoted by $c(n_i, n_j)$, represents the communication cost of the message. The source node of an edge is called a parent node, while the destination node is called a child node. A node with no parent is called an entry node and a node with no child is called an exit node. The precedence constraints of a DAG dictate that a node can only start execution after it has gathered all of the messages from its parent nodes. An example of DAG, shown in Fig. 1, will be used as an example in the subsequent discussion.

The communication to computation ratio (CCR) of a task graph is a measure of the task graph granularity and can be defined in various ways. We adopted the definition used in [2] which defines CCR as the ratio between the average communication and computation costs in the task graph. *The bottom level* (*b-level*) of a node is defined as the length of the longest path from that node to an exit task. The *top level* (*t-level*) of a node is the length of the longest path from an entry node to that node (excluding the cost of that node). The objective of scheduling is to minimize the schedule length, which defined as $\max_i\{FT(n_i)\}$ across all nodes, by proper allocation of the nodes to the processors and arrangement of execution sequencing of the nodes without violating the precedence constraints. Table 1 summarizes the definitions of the notations used in the paper.

The target system is also assumed to be a fully connected distributed memory multiprocessor system with no regard to link contention and scheduling of message.

3. THE PROPOSED ALGORITHM

In our proposed Algorithm, the number of the processors onto which the nodes scheduled is assumed to be unbounded, which means we can add or delete a processor from the target system dynamically when the algorithm executed. We temporally schedule v nodes of the DAG to v processors with every processor accommodating a node by such an order that the pre-schedule length is minimum. The pre-scheduling order can be proven to be ascent order of nodes' top levels. Therefore the start time of node n_i, $ST(n_i)$, is equal to its t-level, and the pre-schedule length is the length of the *CP* of the DAG. What we are to do next is to transfer the nodes by a proper order, which is similar to a scheduling order in other list-scheduling algorithms, to minimize the schedule length and the number of the employed processors. We consider the CPN-Dominant sequence to be the transference order.

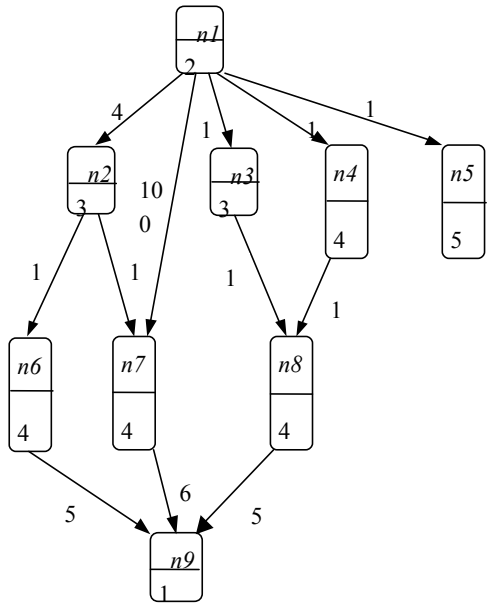


Fig. 1 An Example of Task Graph

In a task graph, there is a set of nodes and edges, forming a path from an entry node to an exit node, of which the sum of computation and communication costs is the maximum. This set of nodes is called the critical path (CP) of the task graph and the CP nodes (CPNs) should be consider for transferring to start as early as possible in the transferring process. However, a CPN can only start its execution if all of its parent nodes have finished their execution. Thus, before a CPN is transferred, all of its parent nodes must be transferred to start execution as early as possible. We call such a node an in-branch node (IBN), which is defined as a node that is not a CPN and from which there is a path reaching a CPN. In addition to CPNs and IBNs, there is another class of nodes called out-branch node (OBN), which are neither CPNs nor IBNs. In terms of this classification, the nodes of any connected graph can be divided into these three categories. Using this categorization, the relative importance of nodes is in the order: CPNs, IBNs, and OBNs. In the CPN-Dominant sequence, the precedence constraints among nodes are preserved in the following manner: the IBNs reaching a CPN are always inserted before the CPN; OBNs are appended to the sequence in a topological order so that a parent OBN is always inserted before its child OBN. The procedure of constructing a CPN-Dominant sequence is described in [3].

Before transferring the nodes among the processors, we must pre-schedule v nodes to v processors with every node impropriate a processors. The start time of every node prescheduled on its processor is equal to its t-level. To compute the t-level of every node, a topology order of the task graph is needed. The goal of transference after pre-scheduling is to schedule the node on another processor that allows the largest reductions in the start time of the node as well as in the number of processors employed. To determine the largest start time reduction, we need to compute the DAT and the start time of the node on each adjacent processor. A node can be scheduled to a processor if the processor has an idle time slot

Table 1

NOTATION	DEFINITION
v	The total number of nodes in the task graph
e	The total number of edges in the task graph
n_i	A node in the parallel program task graph
$w(n_i)$	The computation cost of node n_i
$c(n_i, n_j)$	The communication cost of the directed edge from node n_i to n_j
p	Number of processors
CP	A critical path of the task graph
CPN	Critical path node
$DAT(n_i)$	The possible data available time of node n_i
$ST(n_i)$	The start time of node n_i
$FT(n_i)$	The finish time of node n_i
CCR	Communication-to-computation Ratio
IBN	In-branch node
OBN	Out-branch node

that starts later than the node's DAT (a node's DAT varies with the processor it is scheduled). The following procedure outlines the computation of the start time of a node on a processor.

Computation of ST (n_i, Q):

Precondition: *m nodes* $\{n_{Q_1}, n_{Q_2}, \ldots, n_{Q_m}\}$ *have been scheduled on processor* Q ($m \geq 0$).

1. Check if there exits some k such that:
   $ST(n_{Q_{k+1}}, Q) - \max\{FT(n_{Q_k}, Q), DAT(n_i, Q)\} \geq w(n_i)$
   where $k = 0, \ldots, m, ST(n_{Q_{m+1}}, Q) = \infty,$ and $FT(n_{Q_0}, Q) = 0$.
2. If such k exists, compute $\max\{FT(n_{Q_l}, Q), DAT(n_i, Q)\}$ with l being the smallest k satisfying the above inequality, and return this value as the start time of n_i on processor Q; otherwise, return ∞.

The DAT of a node on a processor is constraint by the finish times of the parent nodes and the message arrival times. If the node under consideration and its parent node are scheduled to the same processor, the message arrival time of this parent node is simply its finish time on the processor (inter-processor communication time is ignored). On the other hand, if the parent node is scheduled to another processor, the message arrival time will be equal to the sum of the parent node's finish time and the communication time between two nodes.

The Node-Transferring Scheduling algorithm is outlined below. In process of the algorithm, we use a dynamic list as the list of utilized processors, of which the nodes may be deleted when the corresponding processor is unoccupied, and a static list as the list of scheduled task node.

Node-Transferring Scheduling Algorithm (NTSA)

Status NTSA (GAG *G***)**

{
Initialize the list of utilized processors and the list of scheduled task nodes;
Pre-schedule the task graph;
Generate the CPN-Dominant Sequence of DAG;
Initialize a pointer (pointing to the first node of the CPN-Dominant Sequence);
for (every node n_x in the CPN-Dominant Sequence)
{
Set the pointer to this node n_x;
Let $ST(n_x)$ be the t-level of n_x.
for (every processor q_y in the list of utilized processors)
{
Compute the start time of n_x on processor q_y, that is, compute $ST(n_x, q_y)$;
if ($ST(n_x, q_y) \le ST(n_x)$)
{
Transfer node n_x to processor q_y from the processor q_x it currently occupies;
if (processor q_x is void now) delete the q_x from the list of the employed processors;
$ST(n_x) = ST(n_x, q_y)$;
$FT(n_x) = ST(n_x) + w(n_x)$;
}
else let n_x be intact;
}
}
$SL = \max_i \{FT(n_i)\}$;
return SL;
}

The pre-scheduling step takes $O(e)$ time because the t-levels of nodes can be computed in $O(e)$ time. The procedure of generating a CPN-Dominant Sequence takes $O(e+v)$ time. The outer for-loop has v iterations and the inner for-loop has v iterations at worst. In the inner loop, it takes $O(e)$ time to compute ST and DAT values of the node on each utilized processor. Thus, the outer for loop takes $O(ev^2)$ time and the NTSA algorithm takes $O(ev^2)$ at worst.

Table 2

node	t-level	b-level	CPN-Dominant order
n_1	0	23	1
n_2	6	15	2
n_3	3	14	5
n_4	3	15	4
n_5	3	5	9
n_6	10	10	7
n_7	12	11	3
n_8	8	10	6
n_9	22	1	8

4. AN ILLUSTRATING EXAMPLE

In this section, we use Fig. 1 for an illustrating example of NTSA algorithm. Table 2 shows the t-levels, b-levels and the CPN-Dominant sequence of the nodes. The critical path of the task graph is $\{n_1, n_7, n_9\}$.

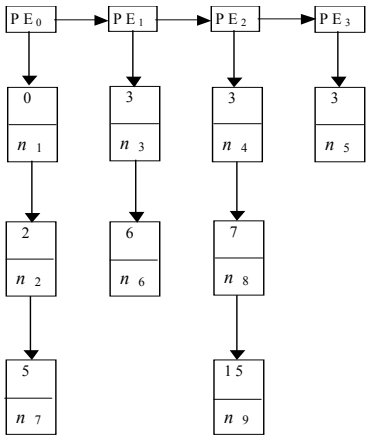


Fig. 2 The result schedule of NTSA

The CPN-Dominant sequence is as follows:

$n_1, n_2, n_7, n_4, n_3, n_8, n_6, n_9, n_5$

The result of NTSA algorithm is shown in Fig. 3. (The number above n_i is the start time of the node on the processor).

5. EXPERIMENT RESULTS

Because of the NP-completeness of scheduling problem, heuristic ideas used in NTSA cannot always lead to an optimal solution. Thus, it is necessary to compare the average performance of different algorithms by using randomly generated graphs, fork-join graphs, in-tree graphs and out-tree graphs. Fig. 4 to Fig.7 shows the result of the comparison of NTSA with famous DCP and DSC algorithms.

6. CONCLUSIONS

In this paper, we have presented a new list-scheduling algorithm for unbounded number of processors, which has good performance compared with other algorithms. The proposed algorithm works better on various types of graph structures with tolerable time complexity. Although, it is devised under the assumption of fully connected network of processors, it can be generalized to networks of other topological structure with the distance of the processors taken into account when the parallel program is complied.

7. REFERENCES

[1] Andrei Radulescu and Arjan J.C. van Gemund, "Low-Cost Task Scheduling for Distributed-Memory Machines", IEEE Transactions on Parallel and Distributed Systems, Vol. 13, No.6, June 2002, pp648-658.

[2] Yu-Kwong Kwok and Ishfaq Ahmad, "Benchmarking and Comparison of the Task Graph Scheduling Algorithms", Journal of Parallel and Distributed Computing 59, 381-422 (1999)

[3] Ishfaq Ahmad and Yu-Kwong Kwok, "On Parallelizing the Multiprocessor Scheduling Problem", IEEE Transactions on Parallel and Distributed System, Vol. 10, No. 4, April 1999, pp414-432.

[4] M. R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Co., 1979.

[5] Min-you Wu and Daniel D. Gajski, "Hypertool: A Programming Aid for Message-Passing Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 1, No. 3, July 1996.

[6] Tao Yang and Apostolos Gerasoulis, "DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors", IEEE Transactions on Parallel and Distributed Systems, Vol. 5, No. 9, September 1994, pp-951-967.

[7] Yu-Kwong Kwok and Ishfaq Ahmad, "Dynamic Critical-Path Scheduling: An Effective Technique for Allocating Task Graphs to Multiprocessors", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No.5, May 1996.

[8] G. -L. Park, B. Shirazi, J. Marquis, and H. Choo, "Decisive Path Scheduling: A New List Scheduling Method", Proc. Int'l Conf. Parallel processing (ICPP), Aug. 1997, pp472-480.

[9] M. A. Palis, J.-C. Liou, and D.S.L. Wei, "Task Clustering and Scheduling for Distributed Memory Parallel Architectures", IEEE Transactions on Parallel and Distributed Systems, Vol. 7, No.1, January 1996, pp46-55.

[10] S.J. Kim and J.C. Browne, "A General Approach to Mapping of Parallel Computation Upon Multiprocessor Architectures", Proc. Int'l Conf. Parallel Processing (ICPP), Vol. 3, Aug. 1988, pp1-8.

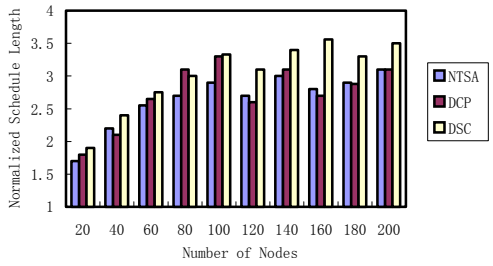


Fig. 4 erage normalized schedule lengths with respect to a group of completely random graph)

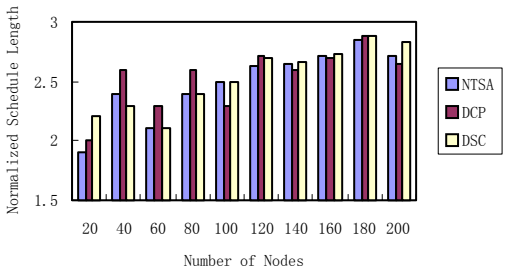


Fig .5 erage normalized schedule lengths (with respect to a group of fork-join graph)

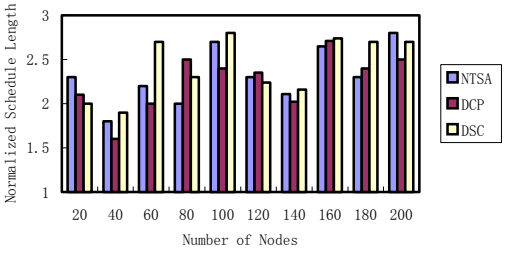


Fig. 6 verage normalized schedule lengths at various graph sizes for in-tree graphs

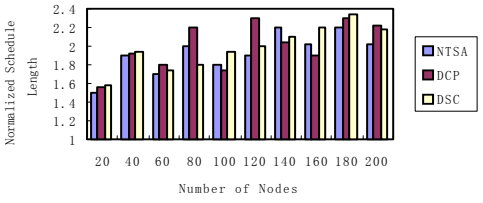


Fig.7 verage normalized schedule lengths at various graph sizes for out-tree grap

A Parallel Discretization Method to Solve an Integral Equation of the First Kind

Wang Weicang Guo Qingping
Wuhan University of Technology, Wuhan 430063, P.R.China
E-mail:Wangweicang@hotmail.com qpguo@public.wh.hb.cn

ABSTRACT

The integral equation of the first kind is the common mathematical model for many inverse problems. In this paper, we propose a parallel algorithm to solve an integral equation of this type. We theoretically prove the convergence of the present parallel algorithm.

Keywords: Discretization, Collocation point, Parallel algorithm, Convergence

1. INTRIDUCTION

In many examples of ill-posed problems, the problem was to solve integral equations of the first kind. Integral operators are compact operators in many natural topologies under very weak conditions on the kernels. The next theorem implies that linear equations of the form $Kx = y$ with compact operators K are always ill-posed.([1])

Theorem 1.1 Let X, Y be normed spaces and $K : X \to Y$ be a linear compact operator with null space $N(K) = \{x \in X \mid Kx = 0\}$
Let the dimension of the factor space $X / N(K)$ be infinite. Then there exists a sequence $(x_n) \subset X$ such that $Kx_n \to 0$ but (x_n) does not converge. We can ever choose (x_n) such that $\|x_n\| \to \infty$. In particular, if K is one-to-one, the inverse $K^{-1} : Y \supset K(X) \to X$ is unbounded.

So this type of equation is the common mathematical model for the inverse problem and therefore the general features of such equations should be kept ever in mind in our coming discussions.

2. TIKHONOV REGULARIZATION

The generalized solution $x = K^{+} y$ of equation

$$Kx = y \tag{1}$$

is the least squares solution and therefore it satisfies the normal equation

$$K^{*}Kx = K^{*}y \tag{2}$$

where K is a compact linear operator from a Hilbert space H_1 into a Hilbret space H_2 and K^{*} is the adjoint of K. Now, the self-adjoint compact operator $K^{*}K$ has nonnegative eigenvalues and therefore, for any positive number α, the operator $K^{*}K + \alpha I$, where I is the identity operator on H_1, has strictly positive eigenvalues. In particular, the operator $K^{*}K + \alpha I$ has a bounded inverse, that is, the problem of solving the equation

$$(K^{*}K + \alpha I)x_{\alpha} = K^{*}y, \tag{3}$$

is well-posed. The second kind equation (3) is called a regularized form of equation (2) and its unique solution

$$x_{\alpha} = (K^{*}K + \alpha I)^{-1} K^{*} y \tag{4}$$

is called the Tikhonov approximation to $K^{+}y$,the minimum norm solution of (2).They converge to $K^{+}y$ as α tends to zero. That is ,

$$\lim_{\alpha \to 0} \left\| x_{\alpha} - K^{+} y \right\|^{2} = 0 .$$

Moreover, since for each fired $\alpha > 0$, the operator $(K^{*}K + \alpha I)^{-1} K^{*}$ is bounded ,we see that the Tikhonov approximation x_{α} depends continuously on y, for each fixed α>0.

To summarize, in Tikhonov regularization, we approximation the minimum norm least square solution $K^{+}y$, which depends discontinuously on y, by a vector x_{α} ,depending on a regularization parameter $\alpha > 0$, which is continuous function of y. To put it another way, an ill-posed problem is approximated by a family of nearby well-posed problems. The details can be consulted in [1].

3. DISCREZATION

For an ill-posed Fredholm integral equation of the first kind

$$\int_0^1 k(s,t)x(t)\mathrm{d}t = y(s), 0 \le s \le 1; \tag{5}$$

we will express this equation abstractly, as we did in the previous section, as an equation of the form (1), where K is the integral operator with kernel $k(s,t)$.

An integral kernel $k(s,t)$ is called degenerate if it has the form

$$k(s,t) = \sum_{j=1}^{n} S_j(s) T_j(t) .$$

If $k(s,t)$ is degenerate, then equation (5) has a solution implies that y lies in the span of $\{S_1, \ldots, S_n\}$.

Suppose that $k(s,t) \in L^2([0,1] \times [0,1])$. If K is the integral operator defined on the real space $L^2[0,1]$ by

$$(Kf)\int_0^1 k(s,t) f(t)\mathrm{d}t ,$$

then the adjoint operator K^{*} of K has the form

$$(K^{*}g)(s) = \int_0^1 k(t,s) g(t)\mathrm{d}t .$$

Now we can suggest a diseretization method in which Tikhonov regularization can be turn into a finite-dimensional problem. In this method the regularization is performed first. When K is the integral operator with kernel $k(s,t)$, $K^{*}K$ is itself an integral operator

$$(K^{*}K)x = \int_0^1 \bar{k}(s,t)x(t)\mathrm{d}t\,, \tag{6}$$

where the kernel $\bar{k}(s,t)$ is given by

$$\bar{k}(s,t) = \int_0^1 k(u,s)k(u,t)\mathrm{d}u\,. \tag{7}$$

Suppose that we apply a quadrature rule to the integral defining the kernel $\bar{k}(s,t)$,

$$\int_0^1 k(u,s)k(u,t)\mathrm{d}u \approx \sum_{j=1}^{n} w_j k(u_j,t)$$

is a finite sum of products of functions of s and t alone, i.e., it is a degenerate kernel. Replacing $K^{*}K$ in (3) with the finite rank operator generated by the degenerate kernel (8), then results in a Fredholm integral equation of the second kind with degenerate kernel. Such an equation is equivalence to a finite dimensional linear system and hence may be solved by algebraic mean.[2]

4. ALGORITHM AND CONVERGENCY

Let < , > be the inner product on $L^2([0,1]\times[0,1])$. If we adopt the approximation of integral kernel as (8), then equation (8) turn to be the form

$$\alpha x_\alpha + \sum_{j=1}^{n} w_j z_j k(u_j,t) = g\,, \tag{9}$$

where

$$z_j = < x_\alpha, k(u_j,s) >, \quad g = K^{*}g.$$

From [2], αx_α has the form

$$\alpha x_\alpha = g - \sum_{j=1}^{n} w_j v_j k(u_j,t)\,, \tag{10}$$

where $v_j(j=1,\cdots,n)$ are unknown coefficients. As we substitute x_α in (10),

we have an equation

$$\sum_{j=1}^{n} w_j v_j k(u_j,t) - \sum_{j=1}^{n}\left(w_j z_j - \frac{1}{\alpha}\sum_{i=1}^{n} w_i w_j v_j c_{ij}\right)k(u_j,t) = 0 \tag{11}$$

where

$$c_{ij} = < k(u_i,t), k(u_j,t) >, i,j = 1,\cdots,n.$$

We can choose collocation points such that $\{k(u_j,t): j=1,\cdots,n\}$ are independent and select the weights $w_j(j=1,\cdots,n)$ not being zero. So we have linear equations

$$v_j + \frac{1}{\alpha}\sum_{i=1}^{n} w_i c_{ij} v_i = z_j,\ j = 1,\cdots,n. \tag{12}$$

They are written as a linear system with matrix form

$$Av = z\,. \tag{13}$$

We obtain the solution of (9) by solving (13)

$$\tilde{x}_\alpha = \frac{g}{\alpha} - \frac{1}{\alpha}\sum_{j=1}^{n} w_j v_j k(u_j,t). \tag{14}$$

For some examples as (3.25) of [1], we may choose collocation points u_j such that the coefficient matrix A is invertible and tridiagonal. Therefore, we can design a parallel algorithm to obtain approximation of (3) as below.

Parallel algorithm:

Step1. Choosing collocation points such that $\{k(u_j,t): j=1,\cdots,n\}$ is independent.

Step2. Choosing weights $w_j(j=1,\cdots,n)$ in (8) being nonzero.

Step3. Computing z_j and c_{ij} to obtain coefficient matrix of (13).

Step4. Blocking coefficient matrix of (13) and dividing (13) into some linear systems with less degree.[4]

Step5. Sending these linear systems to differential processors and iterative solving them at processors respectively.

Step6. Combining results at processors to obtain an approximation $v^{(p)}$ of (13).

Step7. Repeating step5~6 and stopping at correct accuracy.

Let $v^{(p)} = (v_1^p,\cdots,v_n^p)$, then we get an approximation $x_\alpha^p = \frac{g}{\alpha} - \frac{1}{\alpha}\sum_{j=1}^{n} w_j v_j^p k(u_j,t)$ of (9) and x_α^p satisfying

$$\lim_{p\to\infty} \| x_\alpha^p - \tilde{x}_\alpha \| = 0.$$

By choosing nodes, (8) satisfies the more accuracy as n tends to infinite ([3]). Recalling the result of regularization, we have that

$$\lim_{\alpha\to 0} \| x_\alpha - K^{+}y \| = 0.$$

Therefore, as $p\to\infty, n\to\infty, \alpha\to 0$,

$$\| x_\alpha^p - K^{+}y \| \to 0.$$

5. CONCLUSION

We give a discretization method of integral equation of first kind, and design a convergence parallel algorithm to obtain approximation of integral equation. From the previous section, we know that the convergence rate of algorithm is dependent on collocation points. So it is interest for us to find correct collocation points to integral kernels respectively.

6. ACKNOWLEDGEMENT

This work was supported by Natural Science Foundation of China (No.60173046) and Science Foundation of Hubei Province (No.2000J153).

7. REFERENCES

[1] Kirsch,A., An introduction to the mathematical theory of inverse problems, Springer-Verlag, 1996, Berlin.

[2] Riesz.F.,B. Sz.-Nagy, Lecture on functional analysis, 1965, Budapest.

[3] Groetsch,C.W., Inverse problems in the mathematical sciences, 1993,Vieveg.

[4] Guo Qingping et al, Optimize algorithm of multi-grid parallel with virtual boundary forecast, Jour. Nume. and Comp. 2, 2000, Beijing.

Intelligent Job Allocation for Time Constrained Parallel Processing Problems

Purusothaman T
Department of Computer Science and Engineering, Govt. College of Technology
Coimbatore – 641013, India
E-Mail: purushgct@yahoo.com
And
Vijay Ganesh H
Department of Computer Science and Engineering, Govt. College of Technology
Coimbatore – 641013, India
E-Mail: hvg_y2k@yahoo.com
And
Uthra Kumar B
Department of Computer Science and Engineering, Govt. College of Technology,
Coimbatore – 641013, India
E-Mail: uthra_kumar@yahoo.com
And
Chockalingam C T
Department of Computer Science and Engineering, Govt. College of Technology,
Coimbatore – 641013, India
E-Mail: ravi_gct@yahoo.com

ABSTRACT

In this paper we present an Intelligent Job Allocation algorithm for scheduling time constrained parallel processing problems in distributed systems. The key feature of the algorithm is its guarantee of a maximum computational time for the accomplishment of a job. The algorithm obtains a time constraint from the user within which the job is to be completed. Then the algorithm schedules the job among various processing elements aiming at finishing the job in the stipulated time. The algorithm achieves this using an intelligent learning method by storing the previous system performance and performing the current allocation by referring to this history. The previous allocations help obtain a schedule with improving accuracy for every successive job run.

Keywords: Job Allocation, Intelligent Scheduling, metacomputing, grid computing, parallel problems, time constraint.

1. INTRODUCTION

In recent years an increasing number of parallel computers have become a part of so called computational grids or metacomputers [1][2]. Such a grid typically contains many computers offering a variety of resources. Some of the most notable among these research projects include Globus [3], Legion [4] and Condor [5]. The scheduling system is responsible to select these best suitable machines in this grid for user jobs. In large grids it is very cumbersome for an individual user to select these resources manually. The management and scheduling system generates job schedules for each machine in the grid by taking static restrictions and dynamic parameters of jobs and machines into consideration.
The job scheduling for a single parallel computer significantly differs from scheduling in a metacomputer. The scheduler of a parallel machine usually arranges the submitted jobs in order to achieve a high utilization. The task of scheduling for a metacomputer is more complex as many machines are involved with mostly local scheduling policies. The metacomputing scheduler must therefore form a new level of scheduling which is implemented on top of the job schedulers. Also it is likely that a large metacomputer may be subject to more frequent changes as individual resources may join or exit the grid at any time. Note that many users take a special advantage of a computational grid in the potential combination of many resources to solve a single very large problem. This requires the solution of various hardware and software challenges in several areas including scheduling [6].
Though many job allocation algorithms such as the First Come First Serve, Random and Backfill [6] algorithms are commonly available, most of them do not emphasize on the time requirement constraint that is very critical in certain systems. We have come out with an algorithm for generic parallel processing in a real time distributed system. The key feature of the algorithm is that it ensures a tentative schedule for the accomplishment of the job. The schedule changes when processing elements enter or leave the system with the latter occurring when a computer overloads or crashes.
The next section provides a brief description of the basic modules of the algorithm and their functions, while, sections 3 and 4 discuss the load calculator and Job Scheduler modules in detail. The results obtained during test runs are presented in section 5.

2. BACKGROUND

Despite the performance potential that distributed systems offer for resource-intensive parallel applications, actually achieving the user's performance goals can be difficult. One of the most fundamental problems that must be solved to realize good performance is the determination of an efficient schedule. Effective scheduling by the application developer or end-user involves the integration of application-specific and system-specific information, and is dependent on the dynamic interactions between an application and the relevant system(s). Currently, the performance-seeking end-user must develop schedules for distributed heterogeneous applications off-line, using intuition to predict how the application will perform at the time it will execute. The users or application developers must select a configuration of resources based on load and availability, evaluate the potential performance of their application on such configurations (based on their own performance criteria), and interact with the relevant resource management systems in order to implement the application [7].

The allocation process of a scheduler consists of two parts, the selection of the machine and the scheduling over time.

Selection strategies
Selection strategies are strategies for selecting machines for a job request. In the following, M_{max} denotes the machines that are able to execute a specific job in the metacomputer. M_{free} is the subset of machines that have currently enough free resources to start the job immediately [6].

- BiggestFree takes the machine from M_{free} with the largest number of free resources. A disadvantage of this strategy is a possible delay of a wide job, as small jobs may take the critical resources necessary for the next wide job.
- Random chooses a machine from the sets M_{max} or M_{free} by random. On average it provides a fair distinction of the jobs on the available machines.
- BestFit takes the machine either from M_{max} or M_{free} that leaves the least free resources if the job is started. In comparison to BiggestFree this strategy does not unnecessarily fill up larger machines with smaller jobs.
- EqualUtil chooses machine with the lowest utilization to balance the load on all machines [8]. Note that this strategy does not try to keep larger machines free for larger jobs that may be a drawback.

Scheduling Algorithms
Most common algorithms in scheduling are based on list-scheduling. In the following three variants are presented [6, 9]:

- First-come-First-serve: The scheduler starts the jobs in the order of their submission. If not enough resources are currently available, the scheduler waits until the job can be started. The other jobs in the submission queue are stalled. This strategy is known to be inefficient for many workloads as wide jobs waiting for execution can result in unnecessary idle time of some resources.
- Random: The next job to be scheduled is randomly selected among all jobs that are submitted but not yet started, therefore the schedule is non-deterministic. No job is preferred, but jobs submitted earlier have a higher probability to be started before a given time instant.
- Backfill: This is an out-of-order version of FCFS scheduling that tries to prevent the unnecessary idle time caused by wide jobs. Two common variants are EASY and conservative backfilling [10, 11]. In case that a wide job is waiting for execution other jobs can be started under the premise that the wide job is not delayed. Note that the performance of this algorithm relies on a sufficient backlog.

In this paper we have presented a selection strategy algorithm that selects machines for job allocation on an intelligent manner. The main difference between the proposed algorithm and the existing algorithm is the maximum time guarantee that our algorithm is able to provide to the user. For each job that a user submits to the metacomputer, the user gives the maximal computation time constraint within which a job must be completed. Based on this time constraint, the scheduler selects the nodes to which job must be partitioned and allocated. The algorithm is described in detail in the next section.

3. BASIC MODULES OF THE ALGORITHM

The basic design of the job allocation algorithm includes the following modules:

1) Load calculator - calculates the load information on various processing elements across the distributed system to determine the efficient processing elements.

2) Job Scheduler – schedules the job to various individual processing elements depending upon previous performance and the current efficiencies based on the system load.

Process Load Calculator

The process load calculator is the part that obtains the time constraint from the user and finds out the time taken by the individual processing elements to complete the given task. This time along with the user specified time as reference, is used to calculate the load on the individual processing elements. The load average info of an individual system is the measure of the amount of load on the processor. It is used later with the network efficiency to find out the total efficiency of the individual processing elements. The process can be summarized as follows.

a) Take the size of the whole data set provided by the user= n
b) Take a small part (1%) of the whole data set = a
c) Calculate time T_a to execute 'a' part of the whole job on local machine
d) Obtain time constraint (Total time in which the job must be completed) T_n from the user.
e) Calculate the load average info of the individual processing element using T_n & T_a. Load Average Info (LAI) is the measure of the workload of the processor. It is in the scale of 0 to 100. If the processor is busy with many jobs the load average info will be low. It will be inversely proportional to the processor load. The LAI is calculated using the formula

$$LAI = \frac{T_a \times LAI_{\max}}{T_i} \qquad \text{Eq. (1)}$$

where i = 1,2, … n-1.
The ideal node, which has the load average info as 100, is the node that completes the job in the user specified time. The other systems are rated with this system as the reference. Thus we have the quickest processing system with the largest load average info and hence largest efficiency.
f) During Setup, the administrator sets up a table with time estimates for job completion on various processing elements in the whole system having the following fields
   - Load Average Info of the system
   - Network Efficiency of the system
   - Total Efficiency of the system
   - Number of previous runs

   This table is the key feature of the process load calculator. This table is responsible for the intelligent job allocation mechanism with memory of the previous experiences. The table gets updated on every job completion. The table gets more and more accurate with many job runs thus predicting the network conditions and completing jobs in the stipulated time. The administrator sets up the initial table while connecting the processor in the system. This is done by the administrator only during initialization and the table improves by itself once it has been installed. During initialization of any processing element in the distributed system the system administrator runs a sample data in the workstation and notes down the time taken for completion of the job. From this value the load average info is calculated and updated in the table.
g) On every job run the table is updated with the load average info. The load average info is maintained as the average of

all the values obtained so far. This is done by storing the number of previous runs. The other fields get updated in the next part of the algorithm, i.e., in the process scheduler.

The individual processing elements in a distributed system across the network have individual computational efficiencies on the same scale. The systems are selected such that their total efficiency equals the required computational factor. Though the inclusion of all the processing elements in the network will improve the computational factor, it will overload the systems in a long run. So less efficient and overloaded systems are avoided.

Job Scheduler

The job scheduler is responsible for allocating the job fragments to the individual processing elements in the distributed system. The Job Scheduler finds the network efficiency and hence the node efficiency of all nodes. From the efficiencies, the Job Scheduler selects the workstations on which to schedule the job. The whole process can be summarized as follows:

a) Broadcast messages to all individual processing elements and note down the time of broadcast of each message.
b) When a processing element receives the message it responds by relaying its current processor load average info. back to the Job Scheduler.
c) The Job Scheduler receives this information from all the processing elements and puts them in a table.
d) Based on the message timestamps the Job Scheduler calculates the Round Trip Latency (RTL) for each individual processing element and adds this information to the table.
e) The Network Efficiency of the system is the measure of the speed of data transfer to and from the node under consideration based on the relative speeds of all the other nodes in the grid. It is in the scale of 0 to 100. The node in the grid with the longest transfer latency is given a network efficiency of 0 and the local system is given a network efficiency of 100. All other systems are given intermediate values relative to these two nodes. Thus the network efficiency is inversely proportional to the latency time. The network efficiency is calculated as follows:

$$NE = \left(1 - \frac{Lat_i}{Lat_{\max}}\right) \times 100 \qquad \text{Eq. (2)}$$

where Lat_i is the round trip latency of the node and Lat_{max} is the maximum latency of all the nodes in the network. The network efficiencies of all nodes are also entered in the table.
f) The Job Scheduler next calculates the efficiency (η) of each individual processing element based on the following formula: -

$\eta = (1-\alpha)$x Load Average Info + α. Network Efficiency

where α -- Network weightage factor

The network weightage factor α is supplied by the user. It is a measure of the amount of parallelism in the user process. If the process is fine grained then α is more close to 1 and for coarse-grained processes α value tends to 0. This is because in fine grained problems there is a high degree of interaction inside the process. The calculations are highly interdependent. One calculation may differ from other and depend on the result of the other. So more than the processing time involved, the network latency to transfer the data between the individual nodes is important.

For coarse-grained problems the process has less interaction in between its sub modules. The calculations are more or less completely independent of others. All are of the same type and they can execute in parallel with other calculations. Then the network latency is not important and the individual processor load info is important because it would be the factor deciding on the speed of execution.
g) The individual efficiencies are normalized on a scale of 0 to 100. These efficiencies are also added to the table. Next the table is sorted based on the efficiencies of the individual processing elements.
h) The standard deviation of the efficiencies of individual processing elements is found. The systems that have a computational efficiency more deviant from the average are removed and the remaining list is used for job allocation. This is done by calculating the difference between the current node efficiency and the mean efficiency of all the nodes in the grid. If this difference is more than the standard deviation on the negative side the node is removed from consideration for job allocation. This is done because if the difference between the nodes efficiency and the mean efficiency of the grid is high it would be better not to allocate job to that machine as it cannot contribute a significant amount of processing power. This is the step that avoids overloading systems which are already running at full capacity.
i) The Total Computational Factor of the job fixed at 100 is taken into account to determine the number of processing elements to select. The top computers with highest efficiencies are picked up until their sum equals the total efficiency required i.e., 100. Also, the percentage of job to be allocated to each workstation is calculated and stored in a new table called the "Job Table".
j) Finally, the network database table is updated with the new Network Efficiency and new Total Efficiency calculated in this step. This again takes the total number of previous runs into account.

The advantage of this type of load calculation and scheduling is that, by allocating jobs only to the minimum number of processing elements that make up for the required computational factor, the exact amount of computational power is used. This does not overload systems by allocating job fragments to all the processing elements in the system. Since the individual elements are undedicated they work under varying load conditions. Some computers that are already working at high load will be found to contribute a very low computational factor and hence they will not be allocated any job fragments. This type of load calculations is necessary to ensure that the individual elements are not overloaded with parallel jobs.

If network conditions are very prohibitive such that it is not possible to complete the job in the network in the stipulated time, the algorithm notifies the user about this. It then schedules the job to best fulfill the user required constraint.

4. SIMULATION MODEL

The simulation model enumerates the steps, which take place during the load calculation. The simulation system is a grid of workstations with varied capacity and environment. It is an undedicated real time system with N nodes. The inputs are supplied at any one of them.

The effective goal of the load calculator is to compute the efficiency of all the systems in the grid and pick the right set of them to allocate the job. The process Load Calculator and Job Scheduler always runs in the system in which the input is supplied. A distributed database is maintained by the network to train the scheduler in choosing the processing elements in

the long run.

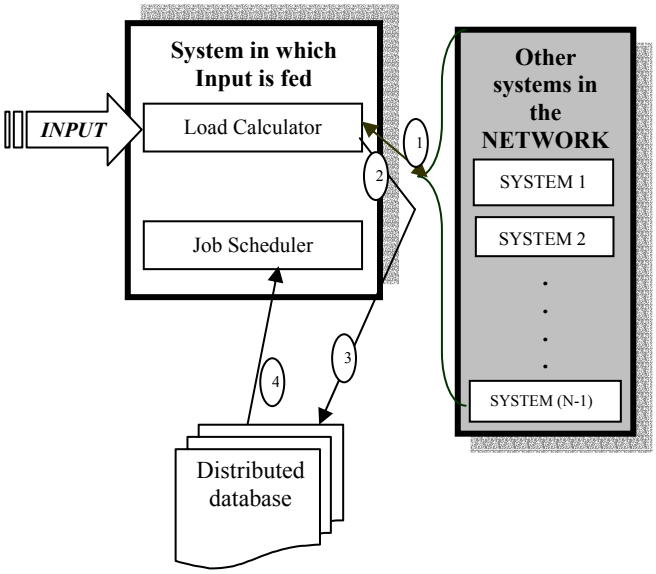


Fig.1 Steps involved in a Load Calculator

Fig.1 shows the steps involved in the load calculator part of the simulation program. The various steps indicated in the figure are described as follows.

INPUT – The input contains the executable file and the data set. This is supplied to only one node in the network.

1 – The Load Calculator module of the program supplies the executable and x% of the data set to the current node and all other nodes in the network. This is established by setting up an individual connection between the current node and each other node in the network.

2 – After executing the part of the data set, the individual nodes respond the Load Calculator with the time taken to execute (T_i where i=1,2,…,N-1,N; N corresponds to the node where the Load Calculator runs). The Round Trip Time (RTT) is calculated later in a similar manner by relaying information between nodes and getting the time involved.

3 – Using the values obtained from the nodes, the Process Load Calculator determines the values for the various fields that are present in the table. The global distributed table maintained by the network is updated using the determined values.

4 – The Job Scheduler uses the values that have been written by the Load Calculator for assigning the efficient nodes.

After the work of the Load Calculator is done, the Job Scheduler will find out the nodes to which the job must be assigned. The steps involved in the Job Scheduler are given as follows.

5 – The Job Scheduler module will consider all the nodes present in the network. It will find out the nodes that are fit to execute the job in hand.

6 – The selection criteria consist of a set of conditions to select the node to execute the job. Initially, the efficiencies of all the nodes are calculated. From the table, the nodes are selected till the sum of individual efficiencies equal the Total Computational Factor (TCF). These nodes are maintained separately as a list and a network connection is established to each of the nodes.

7 – To each node, we send the executable file and the partially split data set that corresponds to the particular node.

5. TEST RESULTS

Two types of problems are considered while executing: - α < 0.5 and α > 0.5.

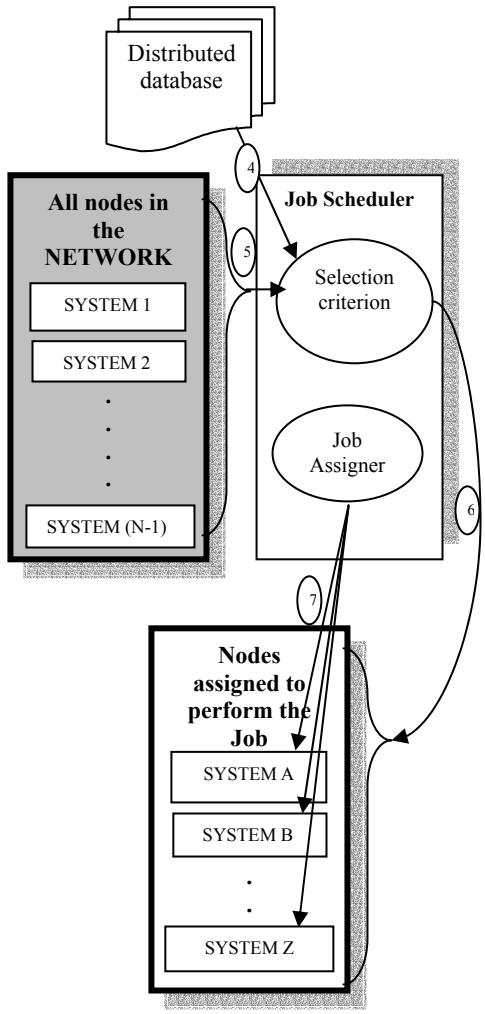


Fig.2 Steps involved in a Job Scheduler

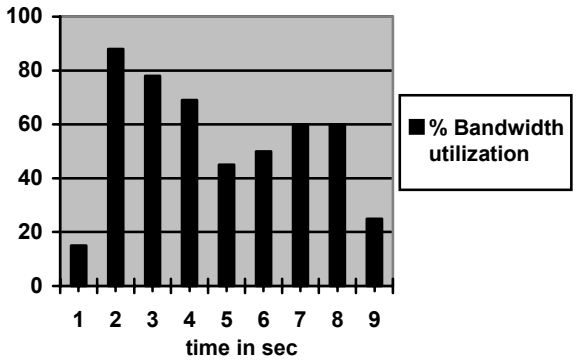


Fig.3 Network Utilization graph for α = 0.85

The peaks in the network utilization graph for the problems with α > 0.5 were almost fully present in the whole time period. The Network weightage factor is of much importance in these types of problems.

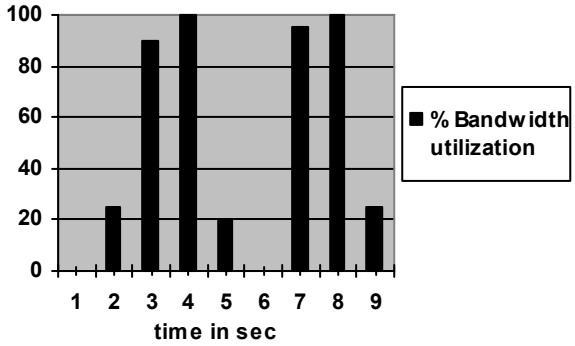


Fig.4 Network Utilization graph for α = 0.32

For problems with $\alpha < 0.5$, the network is utilized only in sending and receiving the data. Processing is done in the remaining time. So the peaks of the Network Utilization graph are concentrated in the initial and the final phases of the time period.

Table 1 Table maintained by Process Calculator and Scheduler

S.No	Time taken to execute (x = 10)% of the input data (msec)	Estimated time to execute 100% of the input data (Ta) (msec)	Round Trip Time (RTT) (msec)
1	260	2600	120
2	310	3100	50
3	220	2200	300
4	250	2500	100
5	350	3500	0

There are several tables maintained during the process of execution of various modules of the program. Some are made global and stored in the distributed table. Some are made private to the module under execution.

Table 2 Distributed Table maintained by the network

S.No	Load Average Info	Network Efficiency	Efficiency	No. of earlier runs
1	58	60	2α+58	10
2	48	83	35α+48	8
3	68	0	-68α+68	11
4	60	67	7α+60	9
5	43	100	57α+43	10

The Process Load Calculator and Scheduler maintain a table with the time for process completion and network transfer latency (Ta and RTT). This is used to compute the Load Average Info and Network Efficiency. These values are used in the Scheduler to calculate the total efficiency of the individual node. This is stored for later reference in a distributed database table as the average of all these values obtained so far with the total number of previous runs.

6. CONCLUSION

As network speeds increase and parallel distributed computing becomes more prevalent, resource-intensive applications will increasingly need to leverage shared, heterogeneous networks of resources. In this paper we presented an algorithm for intelligent job allocation on a grid of heterogeneous workstations. The test results obtained from an implementation of this algorithm have been presented along with the simulation model, which was used for simulation of the algorithm. From the results generated by the simulation, it is clear that Intelligent Job Allocation is effective for coarse grained problems that have high network utilization as well as problems with low network utilization.

7. REFERENCES

[1] European Grid Forum, http://www.egrid.org.
[2] The grid Forum, http://www.gridforum.org.
[3] I. Foster and C. Keeselman. “Globus: A metacomputing Infrastructure Toolkit”. 11(2):115-128, 1997.
[4] A. Grimshaw et al. “Legion: The next logical step towards a nationwide virtual supercomputer”. Technical Report CS-94-21, University of Virginia, Computer Sciences Department, 1994.
[5] M.Litzkow, M. Livny and M. Mutka. “Condor – a hunter of idle workstations”. In Proceedings of the 8th Intl. Conf, on Distributed Computing Systems, pages 104 -111, 1988
[6] Volker Hamscher et al. “Evaluation of Job-Scheduling Strategies for Grid Computing”. In Proceedings of GRID 2000, pages 191-202.
[7] Rich Wolski et al. “Application-Level Scheduling on Distributed Heterogeneous Networks”. UCSD CS Tech Report #CS95-451.
[8] G.D. van Albada et al. “Dynamite- blasting obstacles to parallel cluster computing”. In High-Performance Computing and Networking (HPCN Europe ‘99), Amsterdam, The Netherlands.
[9] J.Krallmann et al “On the design and evaluation of job scheduling algorithms”. In fifth Annual Workshop on Job Scheduling Strategies for Parallel Processing, IPPS ’99; Puerto Rico.
[10] D.G. Feitelson and A.M. Weil. ”Utilization and predictability in Scheduling the IBM SP1 with Backfilling”. In Proceedings of IPPS/SPDP 1998, pages 542-546. IEEE Computer Society, 1998.
[11] D.A. Lifka. “The ANL/IBM SP scheduling system”. IPPS’95 Workshop: Job Scheduling for Parallel Processing pages 295-303.

A New Two Grade Bus-mastering Temperature Detecting System of Computer *

Zhu Jinjun Yang Kuihe Zhao Lingling Zhang Xuemei Zhang Xiaoming
College of Information, Hebei University of Science and Technology, Shijiazhuang Hebei China
E-mail: ykh@hebust.edu.cn

ABSTRACT

This paper introduces a new type two grade bus-mastering temperature detecting system by computer. The bus temperature detecting circuit of temperature intelligence controller is designed, which throws off traditional temperature detecting pattern by use of sensor, translator, multi-switch and A/D converter. This system successfully solves circuitry driving and time cooperating problem of many spots and far distance temperature detecting. The token-ring transferring technique of computer network is skillfully migrated to the temperature detecting system. The paper points out its application and extending value.

Keywords: sensor, bus, temperature detecting.

1. INTRODUCING

The two grade bus-mastering temperature detecting system by computer is designed for temperature detecting of electronic deepfreezes which is on electronic deepfreezes product line. The temperature detecting system can be used to detect the refrigeration capability of electronic deepfreezes. The temperature detecting system adopts bus technique and module structure, realizes distributing detecting system composed by temperature sensor, intelligence controller and main controlling computer . The bus temperature detecting circuit of temperature intelligence controller is designed, which throws off traditional for sensor, translator, multi-switch to A/D converter pattern. This system successfully solves circuitry driving and time cooperating problem of many spots and far distance temperature detecting. The getting rid of dithering the hardware circuit is designed and the token-ring transferring technique of network is skillfully migrated to the temperature detecting system, which set up a new type of very steady and reliable temperature detecting system.
The system has obtained first class prize of Heibei province education bureau science and technology progress and third class prize of Heibei province science and technology progress. The paper points out its application and extending value.

2. THE COMPOSITION OF THE TEMPERATURE DETECTING SYSTEM

The main part of the system is composed of a main controlling computer, 5 intelligence controllers and 400 temperature sensors, as Fig1 shows.
The 5 former grade intelligence controllers of the temperature detecting system separately take charge for temperature data collection of 40 work location's 80 detecting spots and 2 environment location of the group itself. The back grade main controlling computer answers for the work of system control, data stat, temperature curve building, printing report forms and so on. The back grade main controlling computer communicates with 5 former grade intelligence controllers by use of its RS232 connection jack. In order to realize the remote distance communicates, there is a RS232/RS422 connection conversion implement, which is used to connect the 5 former grade intelligence controllers by RS422 bus.
The temperature collecting part adopts DS1820 temperature sensors, whose temperature detecting scope is form －19.5℃ to 99.5℃ . The former grade intelligence controllers of the temperature detecting system only need one I/O line to communicate with a number of DS1820 temperature sensors. The DS1820 temperature sensors is fit to many spots distributing temperature measure and a lots of DS1820 temperature sensors can be hung on the same bus to complete the temperature measure of many spots. The former grade intelligence controller will finish one time of temperature measure via sending out a startup command. The DS1820 temperature sensors put the temperature measure result into own portable memory, so the former grade intelligence controller can receive the temperature measure value of DS1820 temperature sensors by sending out reading memories command. There is severe time restriction to operate DS1820 temperature sensors, and there are mistakes easily in sampling signs as a result of long bus distributing capacitance and diversified electromagnetism infection, so chiefly difficult problem to solve is long bus driving issue.

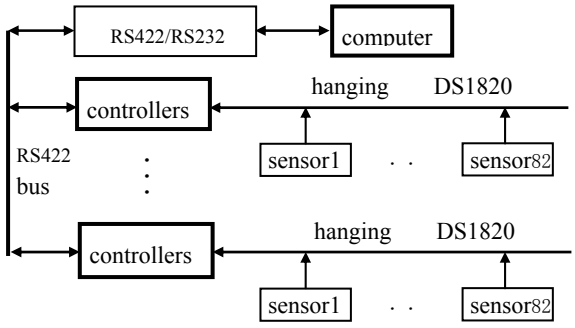


Fig.1 The temperature detecting system fig

3. THE FORMER GRADE INTELLIGENCE CONTROLLERS

3.1 The Composition And Function Of The Intelligence Controller

The hardcore of intelligence controller is 8031 single-chip microcomputer, which controls the work of each components. Program memory saves the controlling program; Data memory saves the temperature measure value of the detecting spot; Coding memory saves the address coding of every detecting temperature exploring head; RS422 connection jack circuit changes the cluster signal into normal RS422 bus signal; Exploring head driving circuit magnifies the controlling signal

* The system has obtained third class prize of Heibei province science and technology progress.

of the single-chip microcomputer; which drives the bus of the detecting temperature exploring head; The watchdog and reposition circuit provides restoration signal for single-chip microcomputer which is produced when the single-chip microcomputer is reset or the single-chip microcomputer program is in disorder as a result of being disturbed accidentally; Number tubes show the work location mark its group, temperature measure value, refrigerating working hours and so on; Keyboard is used to input controlling commands; Besides, the address setting switches is used to set the address of the intelligence controller, which make the 5 intelligence controller have different address.

The intelligence controllers can not only communicate with the main controlling computer, they are but also multi-function controller which can collect temperature independently. Each intelligence controller can work independently and display every spot temperature value of its group and refrigerating working hours; moreover, the intelligence controllers can finish temperature detecting displaying, setting difference in temperature, checking out, startup and stopping controlling of each work spot and so on.

3.2 The hardware technique character of the intelligence controllers

Each DS1820 temperature sensor is of different address coding, and the same bus allows a lots of DS1820 temperature sensors to hang for completing the temperature measure of many spots. The intelligence controller will give off specifically address when it reads and writes to the DS1820 temperature sensors; meanwhile, all the DS1820 temperature sensors on the bus take over the address, but only the DS1820 temperature sensor which has same address obtains the answer, so the CPU can identify the different detecting temperature exploring heads via sending different address.

Because the operating time of DS1820 temperature sensor is fixed, and many controlling impulses is so narrow that they are only a few microseconds long, although almost infinite DS1820 temperature sensors can be hung on the same bus in theory, when the bus is long over 20 meter and the DS1820 temperature sensors are more over 24 in practice, the wave distortion and operating mistakes will be produced as a result of the bus distributing capacitance augmentation. The hardware bus driving circuit of the DS1820 temperature sensors is designed in order to solve this question. The hardware bus driving circuit makes the distance of temperature detecting over 500 meter; 32 cushion stations can be hung on the bus and each cushion station can hang not more than 20 meter connecting line and not more than 24 DS1820 temperature sensors; therefore, a measure temperature bus can connect as many as 500 detecting temperature exploring heads.

3.3 The Software Technique Character Of The Intelligence Controllers

Because the DS1820 temperature sensor connects the intelligence controller through a I/O line; accordingly, the intelligence controller will send out commands and read the temperature data via I/O line, the operating time of the DS1820 temperature sensor is the key question when the program is write. This program adopts C51 computer language and A51 computer language to write together. C51 computer language is simple and is exploited fast, so C51 computer language is adopted in the place where the time is not required strictly; instead, A51 computer language of easily controlling time is adopted to operate the DS1820 temperature sensor, including sending out resetting and command, reading data, etc.

The program skillfully migrates token-ring transferring technique of computer network to the temperature detecting system, adopts simile as similar as token-ring transferring technique of computer network to send out startup command to DS1820 temperature sensor to finish measuring temperature once, and keep detecting time and DS1820 temperature sensor operating time synchronously by program which has solve the stable and dependable information sampling problem. The program divides the time into 50 ms time slices, operating DS1820 temperature sensor once each 50 ms time slices, for example, starting all DS1820 temperature sensors, reading first way temperature values, reading second way temperature values and so on. This process takes not more than 5 ms; therefore, the rest time is used to check whether the key is pressed. If a key is pressed, the corresponding manage is done; or the displaying is renewed. Besides, the program should respond cluster jack break(communicating with main controlling computer) and 50 ms timer break. In addition, the program can finish long line over loading prompting, inserting exploring heads on line, registering automatically, etc. The whole program is very big, including many modules, with complex program flowing. For briefness, fig 2 shows the simple program flowing.

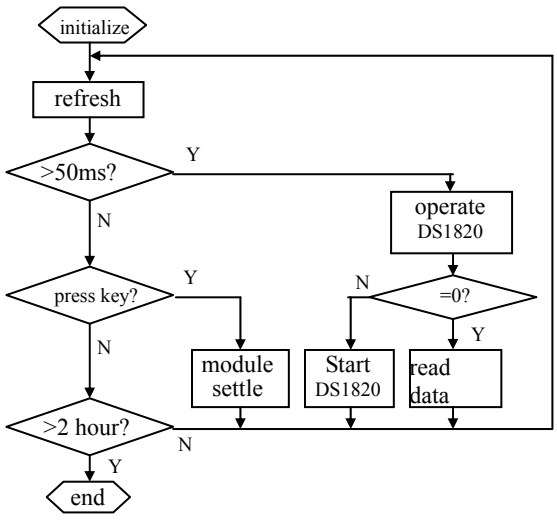


Fig.2 The program flowing of intelligence controller

4. THE MAIN CONTROLLING COMPUTER SYSTEM

The back grade main controlling computer is responsible for reading data from intelligence controller, data analyzing and stating, printing report forms and so on. Also, the back grade main controlling computer can get temperature values from intelligence controller at real time and finish data management at background. The main controlling computer can track the temperature collection of all the electronic deepfreezes in entire process, print the dropping temperature characteristic curve for every electronic deepfreezes, and calculate whether the electronic deepfreeze is up to grade according to temperature dropping value of each detecting spot and standard temperature difference which is set by the system. The system writes oriented object program with VB6.0, which make the display interface beauty and straight.

Fig 3 shows the dropping characteristic curve of a electronic deepfreeze in two temperature detecting spots.

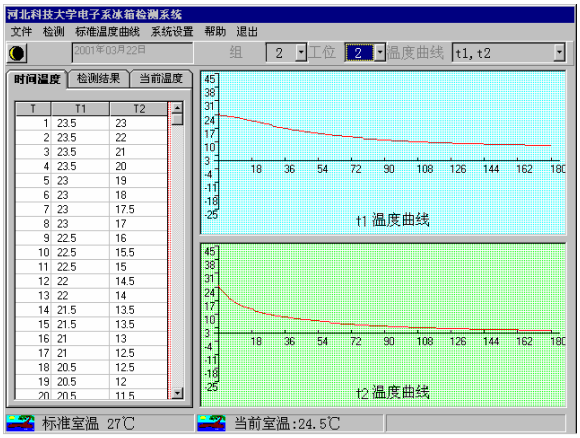


Fig.3 the dropping characteristic curve in real time

5 . THE SUMMARIZING

In conclusion, The two grade bus-mastering temperature detecting system by computer is advanced in technique, with reasonable structure and strong expansibility. Its capability is steady and dependable since it has been used in Heibei province saving energy investing corporation from Mar 1999. The system is appraised by province level in Dec 2000; besides, the system has obtained first class prize of Heibei province education bureau science and technology progress in May 2000 and third prize of Heibei province science and technology progress in Sep 2000. The two grade bus-mastering temperature detecting system by computer has bring obvious economy and society benefit, and has been extended beer factory product line of Luquan city, Heibei province.

6. REFERENCES

[1] Guan Jian, He You, Ping Yingning. Multi-sensor distributing detecting summarizing. Systems engineering and electronics, Vol.22,No.12, 2000,pp11-15.

[2] Li Jifeng, Fu Qiang. The distributing temperature detecting network design based on CAN bus, Micro-computer information, Vol.15,No.6, 1999,pp6-8.

[3] Sun Xianyi, Wang Zhixiao, An Weirong. Anti-jamming problem of long distance temperature measure. North Jiaotong university transaction, Vol.21,No.3, 1997,pp355-358.

[4] Zhao Danfeng, Liu Xi. The use of integration temperature sensor many spots temperature measure. Sensor technology, Vol.16,No.1, 1997,pp36-38.

[5] Sun Ao, He Xiwen, Xu Chengshen,Chen. The expert network for factory automation using multi-sensor information fusion. First Information Conference on Multisource-Multisensor Information Fusion, Las Vegas, USA, 1998.

[6] Zhang Z Y, Grattan K T, Palmer A W. A novel signal processing scheme for a fluorescence based on fiber-optic temperature sensor. Review of Scientific Instruments, Vol.62,No.7, 1991,pp1735-1742.

Dynamic Scheduling Algorithm for Real-Time Applications in Grid Environment

Lichen Zhang
Department of Computer Science and Technology
Guangdong University of Technology
510090 Guangdong, China
E-mail: lchzhang@gdut.edu.cn

ABSTRACT

Advances in networking infrastructure have led to the development of a new type of "computational grid" infrastructure that provides predictable, consistent and uniform access to geographically distributed resources such as computers, data repositories, scientific instruments, and advanced display devices . Such Grid environments are being used to construct sophisticated, performance-sensitive applications in such areas as supercomputer-enhanced instruments, desktop supercomputing, tele-immersive environments, and distributed super computing. Such applications designed to execute on "computational grids" frequently require dynamic scheduling of multiple resources in order to meet performance requirements. Motivated by these concerns, we have developed a general scheduling algorithm in Grid environment. In this paper, we propose a three level dynamic scheduling method. A set of thresholds and information list are used to collect the information of the grid. In first level, as task arrive in the one node of the system. the scheduler uses the information about the task and the state of node , and attempts to guarantee the new task by the execution of this node. If this task can be scheduled on this node and the time constraints can be met, we execute this task on this node. If this task cannot scheduled, the second level scheduling is started., we use the information list of the cluster that is connected closely by the node to find whether this cluster can accept this new task and meet its time constraint. If the cluster can accept this new task, we transfer this new task to one underloaded node of this cluster and executes this new task on this underloaded nodded. If new task cannot be accepted by the cluster closely connected by the node, the third level scheduling is started., we use the information list to find a remote cluster in the grid to accept this task and execute it on the remote cluster in the grid.

1. INTRODUCTION

As the associated human community, instruments, and resources required for data processing become increasingly distributed, real-time online instrument systems connected by wide area networks will be the norm for scientific, medical, and similar data-generating systems. Such systems have rigorous Quality of Service (QoS) objectives. They must behave in a dependable manner, must respond to threats in a timely fashion and must provide continuous availability, even within hazardous and unknown environments. Furthermore, resources should be utilized in an efficient manner, and scalability must be provided to address the ever-increasing complexity of scenarios that confront such systems. The difficulties in engineering such systems arise from several phenomena, one of the most perplexing being the dynamic environments in which they must function. Systems which operate in dynamic environments may have unknown worst-case scenarios, may have large variances in the sizes of the data and event sets that they process (and thus, have large variances in execution latencies and resource requirements), and cannot be characterized (accurately) by constants, by intervals or even by time-invariant statistical distributions. This environment gives rise to the need for a variety of capabilities: dynamically schedulable resources, easily administered and enforced use conditions and access control for all elements, systems designed to adapt to varying conditions in the distributed environment, automated control and guidance systems that facilitate remote (in time, space and scale) operations, and a myriad of reservation and scheduling capabilities for all of the resources involved.

Advances in networking infrastructure have led to the development of a new type of "computational grid" [1][2][16][17] in frastructure that provides predictable, consistent and uniform access to geographically distributed resources such as computers, data repositories, scientific instruments, and advanced display devices . Such Grid environments are being used to construct sophisticated, performance-sensitive applications in such areas as supercomputer-enhanced instruments, desktop supercomputing, tele-immersive environments, and distributed super computing.

Recent advances in real-time systems technology have given us many good schemes for scheduling hard real-time applications[5][6][8][9][21][22]. A weakness shared by most existing schemes is that schedulability of each application can be determined only by analyzing all applications in the system together, i.e., by a global schedulability analysis. While the necessity of global schedulabiiity analysis imposes no serious problem when the system is closed, it does so when the system is open. Here, by ***closed system, we*** mean one in which detailed timing attributes of all real-time applications on each processor are known. Often times, the applications are developed together, and the schedulability of every combination of applications that can run at the same time is determined a ***priori.*** In contrast, in a grid ***environment,*** applications may be developed and validated independently. During run-time, the user may request the start of a real-time application whose schedulability has not been analyzed together with currently executing applications. The system must determine whether to accept the request and admit the new application. It usually admits a new real-time application only when the application and all the existing real-time applications are schedulable. A global schedulability analysis as an acceptance test is sometimes not feasible because many characteristics of real-time applications in the system are unknown. Even when it is feasible, such an acceptance test can be time consuming when the applications are multi-threaded and complex.

In this paper, we propose a three level dynamic scheduling method. A set of thresholds and information list are used to collect the information of the grid. In first level, as task arrive in the one node of the system. the scheduler uses the

information about the task and the state of node , and attempts to guarantee the new task by the execution of this node. If this task can be scheduled on this node and the time constraints can be met, we execute this task on this node. If this task cannot scheduled, the second level scheduling is started., we use the information list of the cluster that is connected closely by the node to find whether this cluster can accept this new task and meet its time constraint. If the cluster can accept this new task, we transfer this new task to one underloaded node of this cluster and executes this new task on this underloaded nodded. If new task cannot be accepted by the cluster closely connected by the node, the third level scheduling is started., we use the information list to find a remote cluster in the grid to accept this task and execute it on the remote cluster in the grid.

2. GRID ARCHITECTURE

Grid Architecture comprises four general types of componts[1][13]: Grid Fabric, Grid middleware, Grid developments and Tools and Grid application and portals. There are three main issues that characterize computational grids:

1.**Heterogeneity:** a grid involves a multiplicity of resources that are heterogeneous in nature and might span numerous administrative domains across wide geographical distances.

2.**Scalability**: A grid might grow from few resources to millions. This raises the problem of potential performance degradation as a Grids size increases. Consequently, applications that require a large number of geographically located resources must be designed to be extremely latency tolerant.

3.**Dynamicity or Adaptability:** in a grid, a resource failure is the rule, not the exception. In fact, with so many resources in a Grid, the probability of some resource failing is naturally high. The resource managers or applications must tailor their behaviour dynamically so as to extract the maximum performance.

The steps necessary to realize a computational grid include:

- The integration of individual software and hardware component into a combined networked resource.
- The implementation of middleware to provide a transparent view of the resources available.
- The development of tools that allows management and control of grid applications and infrastructure.
- The development and optimization of distributed applications to take advantage of the resources.

The componts that are necessary to form a grid are following:

- Grid Fabric: it comprises resources-specific and site-specific mechanisms which, when in place, enable or facilitate the creation of higher-level distributed "Grid Services." Examples of Fabric mechanisms might include network quality of service support in routers and end systems; resource management interfaces supporting advance reservation, allocation, monitoring, and control of computers, storage systems, etc.; instrumentation interfaces; high-speed network interfaces; and specific implementations of network protocols. These Grid Fabric mechanisms serve to Grid-enable basic resources, augmenting or complementing the basic communication functionality provide by the Internet Protocol suit to allow, for example, the coordinated allocation of computers, networks, and storage systems.
- Grid Services or Middleware: it offers core services such as remote process management, co-allocation of resources, storage access, information security, authentication, and Quality of Service (QoS) such as resource reservation and trading.
- Grid Development Environments and Tools: These offer high-level services that allows programmers to develop applications and brokers that act as user agents that can manage or schedule computations across global resources.
- Grid Applications and Portals: They are developed using grid-enabled languages such as HPC++, and message-passing systems such as MPI. Specific Gird-aware application are implemented in terms of various Application Toolkit components, Grid Services, and Grid Fabric mechanisms.

The management of processor time, memory, network, storage, and other component in a grid is clearly very important. The overall aim is to efficiently and effectively schedule the applications that need to utilize the available resources in the metacomputing environment [14]. From a user's point of view, resource management and scheduling should be transparent; their interaction with it being confined to a manipulating mechanism for submitting their application. It is important in a grid that a resource management and scheduling service can interact with those that may be installed locally. The architectural model of resource management systems is influenced by the way the scheduler is structured. The structure of scheduler depends on the number of resources on which jobs and computations are scheduled, and the domain in which resources are located. Primarily, there are three different models for structuring schedulers:

- Centralized scheduling model: This can be used for managing single or multiple resources located either in a single or multiple domains. It can only support uniform policy and suits well for cluster management systems such as Condor, LSF, and Condine. It is not suitable for grid resource management systems as they are expected to honor policies imposed by resource owners.
- Decentralized scheduling model: In this model schedulers interact among themselves in order to decide which resource should be applied to the jobs being executed. In this scheme, there is no central leader responsible for scheduling, hence this model appears to be highly scalable and fault-tolerant. As resource owners can define the policy that schedulers can enforce, the decentralized scheme suits grid systems. Because the status of remote jobs and resources is not available at single location, the generation of highly optimal schedule is questionable. This model seems difficult to implement in the grid environment, as domain resource owners do not agree on a global policy for resource management.
- Hierarchical scheduling model: This model fits for grid systems at it allows remote resource owners to enforce their own policy on external users. This model looks like a hybrid model, but appears more like centralized model and therefore suits grid.

A idea grid environment will therefore provide access to the available resources in a seamless manner such that physical discontinuities such as differences between platforms, network protocols, and administrative boundaries become completely transparent. In essence, the grid middleware turns a radically heterogeneous environment into a virtual homogeneous one.

With the recent adoption of the CORBA Component Model (CCM) [12] application designers now have a standard way to implement, manage, configure, and deploy components that implement and integrate CORBA services. CCM standard not only enables greater software reuse for servers, it also provides greater flexibility for dynamic configuration of CORBA application. Thus, CCM appears to be well –suited for real-time applications based on the Computing Grid.

Meeting the QoS requirements of distributed real-time applications requires an integrated architecture that can deliver

end-end QoS support at multiple levels in real-time and embedded systems. Distributed object computing (DOC) middleware based on the real-time CORBA (RT_CORBA) [12] offers solutions to some resource management challenges and developers of real-time systems, particularly those systems are designed using dynamic scheduling techniques. The OMG has formed a Special Interest Group (RT SIG) [23] with the goal of extending the CORBA standard with real-time extensions. One of the requirements that has been established by the RT SIG involves providing global real-time scheduling to support the enforcement of end-to-end timing constraints on client server interactions. The specify requirements for scheduling in real-time CORBA involve the need for a global priority that has meaning across the entire distributed systems. A scheduling service in a real-time CORBA systems must provide this global priority, based on the timing constraints expressed by the client's method invocations, and on the server's own timing requirements. The service must also ensure that the global priority can be mapped to the scheduling environments of all local operating systems involved in the execution. Finally if the real-time CORBA systems is a dynamic systems, the scheduling service must ensure that the global priority correctly reflects the requirements of the associated execution for the duration of the execution. That is, it may be necessary to modify the value of the global priority based on changes that occur in the system.

The Dynamic Real-Time CORBA system provides the groundwork for a full dynamic real-time CORBA design. However, there is still much work to be done. A number of dynamic scheduling algorithms have been proposed, RT-CORBA must be enforced with them.

3. DYNAMIC SCHEDULING ALGORITHMS

Dynamic scheduling [3][5][18][19][20][23] in real-time systems involves dynamically making a sequence of decisions concerning the assignment of system resources to real-time tasks. System resources include processors, memory, and shared data structures. Tasks may have arbitrary time constraints, different important levels, and fault tolerance requirements. Unfortunately, making these scheduling decisions is difficult, partly because the decisions must be made without the full knowledge of the future arrivals of tasks and partly because scheduling has to deal with many complex issues, e.g.,, multiprocessors and fault tolerance.

Ramamritham's algorithm[23]

Ramamritham et al. proposed combining local and global scheduling approaches in distributed systems for both periodic tasks and aperiodic tasks. In their approach, periodic tasks are assumed to be known a priori and can always be scheduled locally. On the other hand, an aperiodic task may arrive at a node at any time and will be scheduled locally on the node if its deadline can be met there; otherwise, the task will be transferred toa romote node, which was called global scheduling. If non of the remote nodes can guarantee the deadline of this aperiodic task, it will be rejected and may seriously affect the system performance. Hence, the main effort in [23] was to design a heuristic, global scheduling policy so as to reduce the number of rejected aperiodic tasks. Three algorithms, were used to select a remote node for each aperiodic task which cannot be guarantee an aperiodic task locally, it will attempt to locate a remote node which can guarantee the task.

Xu's algorithm [22]

J.Xu proposed a approach that integrates run-time scheduling with pre-run-time scheduling, for the scheduling of both periodic and asynchronous processes with hard or soft deadlines, and different a priori knowledge of the process characteristics. A guiding principle for this approach is that the scheduling algorithm should exploit to a maximum extent of knowledge about system processes characteristics that are available to the scheduler both before run-time and during run-time.

Deng's algorithm[21]

Z.deng and J.W.S.Liu Proposed a two-level hierarchical scheme for scheduling an open system of multi-threaded, real-time applications on a single processor. It allows different applications to be scheduled according to different scheduling algorithms.

Chang's Algorithm[19]

H.Y. Chang proposed a dynamic scheduling algorithm for a soft real-time system. The algorithm has a two phase polling strategy and is based on the "Shortest Processing Time First' local scheduling policy. Unlike hard real-time systems in which a late job may entail catastrophic outcomes, a soft real-time system functions correctly as long as the deadline miss ration and the expected lateness are below pre-defined levels.

Shin's algorithm[20]

In the paper [20], K.G. Shin and Y. C. Chang proposed a load sharing method with state-change broadcast (LSMSCB) for distributed real-time systems, in which each node maintains state information of only a small set of nodes in its physical proximity, called a buddy set. The load state of a node is defined by three thresholds: $TH_{u,}$ TH_f and TH_v. A node is said to be underloaded if its queue length (QL) is less than or equal to TH_u, *medium-loaded*: if $TH_u < QL \leqslant TH_f$, f*ully load*: if $TH_f < QL \leqslant TH_v$, *overloaded:* if $QL > TH_v$. When a node becomes fully loaded (underloaded) duo to the arrival and/or transfer (completion) of tasks, it will broadcast its change of state to all the other nodes in its buddy set. Every node that receives this information will update its state information by eliminating the fully loaded node from, or adding the underloaded node to, its ordered list (called a preferred list) of available receivers. An overloaded node can select, without probing other nodes, the first underloaded node in its preferred list and transfer a task to that node. Moreover, the buddy set of the nodes in one buddy set are different but are not disjoint, thus allowing the surplus tasks in a buddy set to be transferred to many different buddy sets, i.e., system-wide load sharing. As a result, this method is shown to enable tasks to be completed before their deadlines with much higher probability than other known methods.

Corsaro's algorithm[24]

In open distributed real-time and embedded (DRE) systems, different ORB endsystems may use different scheduling policies. To ensure appropriate end-to-end application behavior in an open architecture, however, DRE systems must enfore an ordering on activities originating in an endsystems and activites that migrate there, based on the relative importance of these activities. This paper describes the meta-programming techniques applied in Juno, which is an extension to Real-Time CORBA that enhances the openness of DRE systems with respect to thire scheduling policies by enabling dynamic ordering of priority equivalence classes.

Foster's algorithm[25]

I.Foster proposed an approach to QoS that combines features of both reservations and adaptation to enhancing the end-to-end performance of network applications. At the core of this approach is a QoS architecture in which resources are enhanced with:
Online control interfaces that allow applications, or agents acting on their behalf, to modify resource characteristics dynamically;
Sensors that allow applications to detect when adaptation is required; and
Decision procedures that support the expression of a rich set of resource management policies.

4. DYNAMIC REAL-TIME SCHEDULING IN GRID ENVIRONMENT

Our model of a Grid is as follows. The grid consists of M clusters connected by Internet. Each cluster consists N_i nodes which a communication network. The network is assumed to be logically connected in that every node can communicate with every other node. One node in Cluster M_k can communicate another node in cluster M_j by server k in cluster M_k and server j in cluster M_j. A stream of jobs is submitted locally to node k. We assume that the nodes are heterogeneous in the sense that each node may have a different arrival rate of externally submitted jobs, but homogeneous in the sense that a job submitted at any node in the cluster k in the Grid can be processed at any other node in the cluster k. We also assume that a job on one node of cluster k can be sent to another node of cluster j to process. We consider a real-time system in which a job is lost if it cannot started within a given time constraint, i.e., within a fixed time after its arrival. If a constraint of the job on one node in cluster k cannot be met locally, however, it may transfer it to another node in cluster k if a underloaded node can be found in cluster k. If a underload node cannot be find in cluster k, it may transfer it to another node in cluster j if cluster j is underloded.
In dynamic scheduling algorithms, there are five separable and yet integrated components:
(1) Local scheduling determines the sequencing of local job
(2) Information policy dictates the methods of exchanging load status among nodes and clusters in the Grid.
(3) Initial policy decide when to initiate a migration request.
(4) Candidate selection policy determines how to choose a job for migration.
(5) Location policy defines the terms under which two processors may transfer a job.

In this paper, we proposed a dynamic scheduling algorithms based on Kang G. Shin's method [20]. Each cluster maintains state information of only a small set of clusters in its physical proximity, called a buddy set. The load state of a cluster is defined by three thresholds: CTH_u, CTH_f and CTH_v. There are four states for each cluster:

- ***underloded:*** A cluster is said to be underloaded if its number of jobs (tasks) (N_c) is less than or equal to CTH_u,
- ***medium-loaded***: if $CTH_u < N_c \leqslant CTH_f$
- f***ully load***: if $CTH_f < N_c \leqslant CTH_v$
- ***overloaded:*** if $N_c > CTH_v$.

For each node in any cluster, The load state of a node is also defined by three thresholds: NTH_u, NTH_f and NTH_v.

- ***underloded:*** A node is said to be underloaded if its number of jobs (tasks) (N_n) is less than or equal to NTH_u,
- ***medium-loaded***: if $NTH_u < N_n \leqslant NTH_f$
- f***ully load***: if $NTH_f < N_n \leqslant NTH_v$
- ***overloaded:*** if $N_n > NTH_v$.

Each cluster has one server that can collect the state information of other clusters and the state information of each node of this cluster. When a node becomes fully loaded (underloaded), it will send its change of state to the server, and server of the cluster will update its state information table. If the cluster changes his state, e.g, from underload to fully loaded, it will broadcast its change of state to all the other clusters in its buddy set. Every cluster that receives this information will update its state information by eliminating the fully loaded cluster from, or adding the underloaded cluster to, its ordered list (called a preferred list) of available receivers. Thus dynamic scheduling method is organized at three level. In first level (node scheduling), as task arrive in the one node of the system. The scheduler uses the information about the task and the state of node , and attempts to guarantee the new task by the execution of this node. If this task can be scheduled on this node and the time constraints can be met, we execute this task on this node. If this task cannot scheduled, the second level scheduling (cluster scheduling) is started., we use the state information table of the cluster that is connected closely by the node to find whether this cluster can accept this new task and meet its time constraint. If the cluster can accept this new task, we transfer this new task to one underloaded node of this cluster and executes this new task on this underloaded node. If new task cannot be accepted by this cluster, the third level scheduling (grid scheduling) is started., we use the information list to find a remote cluster in the grid to accept this task and execute it on the remote cluster in the grid.

Algorithm

Initiation:

- Creat the thresholds CTH_u, CTH_f and CTH_v on the server of each cluster
- Creat the state information table , buddy set and preferred list on the server of each cluster
- Creat the thresholds NTH_u, NTH_f and NTH_v on each node

Node Sheduling

```
While a task arrive on one node in a cluster
do
    if this node is idle
        then
            this task will be executed immediately
            send a message to the server to update
              the state information table.
        else
            If QL>NTH_v
              then
                  goto cluster scheduling
              else
                  make this task in queue
                  send a message to the server to
                  update state information table.
While a task finished his execution
do
    if queue of this node is not idle
        then
            task in the head of queue will be
              executed immediately
            send a message to the server to update
            state information table
        else
            send a message to the server to update
            the state information table.
```

Cluster Scheduling

```
If a message to update information arrives
    then
        update state information table.
If the state of cluster changes,
    then
        goto grid scheduling.
If overloaded task is transferred to the server
    then
        if find a underloaded node in local cluster
          then
        migrate overloaded task to this Node to
        execution, update state information table
  else
        goto grid scheduling
```

Grid Scheduling

```
If the state of cluster changes,
    then
        broadcast its change of state to all other clusters
        in its buddy set.
If this cluster is overloaded,
    then
        transfer to overloaded tasks to the first
        underloaded cluster in its preferred list and
        transfer a task to that cluster.
If one overloaded task from other clusters arrives,
    Then
        goto cluster scheduling
```

Since communication cost must be kept below a given required level while minimize the resultant overhead , the main issues of the scheduling algorithms are how to define state of each node and state of each cluster, how to collect state information, and how to redistribute loads among nodes and clusters, such that overloaded nodes will locate underloaded nodes to share their loads in local cluster or remote cluster with a very high probability. Buddy sets, preferred lists, state information table, and threshold patterns are the most important features to resolve these issues.

The buddy set of a cluster is a set of clusters in its physical proximity. Since state information for different clusters is exchanged only within a buddy set and since a constant buddy set size of 10 to 115 nodes is shown to work well regardless of the system size, the communication overhead is reduced to a constant from O(N2), as compared to the case when state information is exchanged in entire systems of N clusters. In order to avoid more than one overloaded cluster "dumping" their loads on one underloaded cluster or surplus tasks being confined in a certain region, the clusters in a buddy set are ordered into a preferred list such that each cluster will be selected as the k_{th} preferred cluster by one and only one other cluster. I t has been shown that the preferred lists can effectively solve both the coordination and congestion problems, thus meeting task deadlines with a high probability.

The exact analysis of scheduling algorithm is difficult. The composite task arrival process at a node or a server of cluster is composed of the local (external) task arrivals and task transfers, the latter of which is itself a composite process of task transfers from different nodes or clusters. One difficulty in estimating the composite task arrival rate is that the transferred-in task arrival process (and thus the composite arrival process) may not be Poisson even if the local task arrival process is Poisson. This is because the probability of sending a task to (or receiving a task from) a node or a cluster depends on the state of both nodes or clusters, making the splitting process non-Poisson, and task transmission times may not be exponentially distributed, making the process of transferred-in tasks non-Poisson. Furthermore, even if we assume the composite arrival process to exhibit behaviors similar to a Poisson process, the transferred-in task arrival rate from a node is not known due to the dynamic change of system state. In this paper, approximate method was used in the algorithm analysis. .Bayesian estimation is used for the on-line computation of the composite task arrival rate on a node. We consider Poisson external task arrivals and we further approximat*e* the composite task arrival process to be Poisson . This approximation rests on a general result of renewal theory which states that the superposition that the arrival rate of tasks with increasingly many component processes yields a Poisson process.

5. CONCLUSION

The Computational Grid provides a promising platform for the efficient execution of complex real-time applications. Scheduling such application is challenging because target resources heterogeneous ,because their load and availability varies dynamically, and because the tasks have the strict timing constraints. In this paper, we proposed an dynamic real-time scheduling algorithm for real-time application running on the Grid. The scheduling model was organized in three levels: node scheduling, cluster scheduling and grid scheduling. This model fits for grid systems and it integrates local scheduling and global scheduling. This model looks like a hybrid model, but suits grid.

6. ACKNOWLEDGMENTS

This work is partly supported by the National Natural Science Fund, Natural science Fund of Guangdong province, "Thousand, Hundred, and ten" outstanding person fund of Education Department of Guangdong Province, Natural science fund of Education Department of Guangdong Province.

7. REFERENCES

[1] Foster, Buliding the Grid: An integrated services and toolkit architecture for next generation networked applications, Http://www.computingportals.org/gce-papers/building_the_grid.htm.

[2] M.Baker and G.Fox, Metacomputing: Harnessing informal supercomputers, High performance cluster computing: Architecture and Systems, R.Buyya (ed.), Volume 1, Prentice Hall PTR, NJ, USA, 1999.

[3] L.R.Welch et al., Specification and modeling of dynamic, distributed real-time systems, Proceedings of the 19th IEEE Real-Time Systems Symposium, 72-81, IEEE Computer Society Press, 1998.

[4] D.C.Schmidt et al., The design and performance of real-time object request brokers, Computer Communication, Vol.21, pp.294-324, Apr.1998.

[5] R.M. Kavi, "Real-time systems: Abstractions, languages, and Design Methodologies", IEEE Computer Society Press, 1992.

[6] J.A. Stankovic and K.Ramarithm, Hard real-time systems, IEEE computer Society, Order Number819, 1988.
[7] S.P. Reiss, 'PECAN: program development systems that support multiple views', IEEE Trans. on Software Eng., SE-11, (3), 1985.
[8] A.M.V. Tilborg and C.M. Koob, Foundations of real-time computing: Formal specifications and methods, Kluwer Academic publishers, 1991.
[9] J.Xu and D.L.Parnas, On statisfying timing constraints in hard real-time systems, Proceeding of the ACM SIGSOFT'91 Conference on Software for Critical Systems, 1991.
[10] L.Zhang, et al., Methodology of real-time system design using multiprocessors, Microprocessors and Microsystems, Vol 17, No 4, May 1993.
[11] L.Zhang and B.Chaib, A design methodology for real-time to be implemented on multiprocessors, the Journal of system and software, April, 1996, 33: 37-56.
[12] L.Dipippo et al., Expressing and enforcing timing constraints in a dynamic Real-Time CORBA system, Real_time Systems, Vol.16, issue 2/3, May 1999.
[13] M. Baker et al., The grid: international efforts in global computing, Internal conference on advances in infrastructure for electronic business, science, and education on the Internet, Italy, 2000.
[14] R. Buyya et al., An architecture for a resource management and scheduling system in a Global computational grid, HPC ASIA'2000, China, IEEE CS Press, USA, 2000.
[15] S.Chapin et al., A grid resource management architecture, Grid Forum scheduling working group, nov. 1999.
[16] J.Dongarra, An overview of computational grids and survey of a few research projects, Symposium on global information processing technology, Japan, 1999.
[17] I. Foster and C. Kesselman, The grid: Blueprint for a new computing infrastructure, Morgan kaufmann publishers, USA, 1999.
[18] H.Casanova et al., Adaptive scheduling for task Farming with Grid middleware, International Journal of Supercomputer applications and high performance computing, 1999.
[19] H.Y.Chang, Distributed scheduling under deadline constaints, a comparision of sender-initiated and receiver-initiated approaches, Proceedings of IEEE Real-Time Systems Symposium, 175-180, IEEE Computer Society Press, 1986.
[20] K.G.Shin and Y.C. Chang, Load sharing in distributed real-time systems with state change broadcast. IEEE Trans. Comput. C-38, 8 (august 1989), 1124-1142.
[21] Z.Deng and J.W.S.Liu, Scheduling real-time application in an open environment, Proceedings of the 18th IEEE Real-Time Systems Symposium, 308-319, IEEE Computer Society Press, 1997.
[22] J.Xu and D.L.Parnas, Integrating run-time scheduling and pre-run-time scheduling of real-time processes, Real-time programming 1998, 73-80, Elsevier Scice Ltd.
[23] Ramamritham et al., Distributed scheduling of tasks with deadlines and resources requirements, IEEE Trans. Comput. C-38, 8 (August 1989), 110-1123.
[24] A.Corsaro et al., Formalizing meta-programming techniques to reconcile heterogeneous scheduling policies in open distributed real-time systems, Proceedings of the 3rd international symposium on distributed objects and application, September 8-10, 2001, Rome, Italy.
[25] I.Foster et al., A distributed resource management architecture that supports advance reservations and co-allocation, Proceedings of International workshop on quality of service, pp.27-36, June 1999.

Distributed Data and System Integration Through Machine Understanding *

Siping Liu, Guozhen Xiao, Qiwei Yin
Institute of Artificial Intelligence, College of Computer Science, Zhejiang University
Hangzhou, Zhejiang 310027, China
E-mail: lsp_zju@sina.com.cn

ABSTRACT

Research on distributed data integration has traditionally focused on the development of systems for the maintenance and interconnection of databases.
In the next few years, public and private organizations will expand their actions to promote the creation of the Semantic Web. It has commonly been accepted that artificial intelligence and data mining techniques may support the interpretation of huge amounts of integrated data. But at the same time, these research disciplines are contributing to the creation of content markup languages and sophisticated programs able to exploit the constraints and preferences of user domains.
This paper discusses a number of issues on intelligent systems for the integration of distributed information resources, and gives a technical framework for this aim in the end.

Keywords: data integration; artificial intelligence; semantic web

1. INTRODUCTION

Research on distributed information integration has traditionally focused on the design of standards to represent and maintain data and knowledge bases and the development of protocols to interconnect heterogeneous and distributed databases. These and more advanced initiatives will provide the core elements for the creation of a global knowledge management infrastructure. This process will not only allow the exchange of information between users, but it will also allow computer programs to automatically search, retrieve and analyze information content on the web. Furthermore, agent-based technologies may become a fundamental knowledge discovery approach [1], in which distributed programs will interact with multiple web resources and humans to improve their cooperation capabilities.
It is well known that artificial intelligence (AI) and data mining techniques are capable of analyzing multiples sources of data, which may be acquired from different computers, represented by different formats and aimed to describe several features of a complex problem domain. But also it has been shown that this knowledge management pattern may significantly support the birth of the Semantic Web mentioned above. Important aspects for the creation of such a Semantic Web include the design of machine-understandable web content and reasoning models [2]. AI-inspired content markup languages are currently being developed and evaluated in domains such as electronic business [3]. These languages and interactive systems will be based on well-defined semantics and processing rules, which will allow a more effective manipulation of web-based resources. However there are several computational factors that deserve further research.
This paper present an overview about some of the problems currently investigated by the computational intelligence community in order to achieve some of the goals of the Semantic Web. Similarly, it discusses some of the potential applications to support the development of breakthrough technologies for distributed data and knowledge integration, and gives a technical framework for this aim in the end.

* The research is supported by National Natural Science Foundation of China (grant no 60174053).

2. THE SEMANTIC WEB ENABLED INTEGRATION

Figure 1 illustrates some of the building modules required in the construction of a distributed intelligent integration infrastructure for the machine-understanding era. A fundamental condition that has already been achieved is the existence of multiple sources of distributed and heterogeneous data. This information (module(0))will need to be represented and stored using standards to facilitate their exchange and analysis. Module (1) in Figure 1 includes some of the techniques used to approach this problem, such as eXtensible Markup Language (XML), XML schemas or document type definitions. A service may be defined as a software platform, which is able to provide users with automatic ways to search, retrieve and analyse information. Module (2) consists of the implementation of services in a machine-understandable form. Another fundamental component is illustrated in module (3). A well-known problem is the difficulty in finding and using the many databases and services currently available on the web. Therefore agent technology has become a promising approach to deal with some of the complexity, reliability and autonomy issues required to support integrative knowledge discovery tasks [4]. The following sections discuss a number of aspects that need to be studied for the development of the platform depicted in Figure 1.

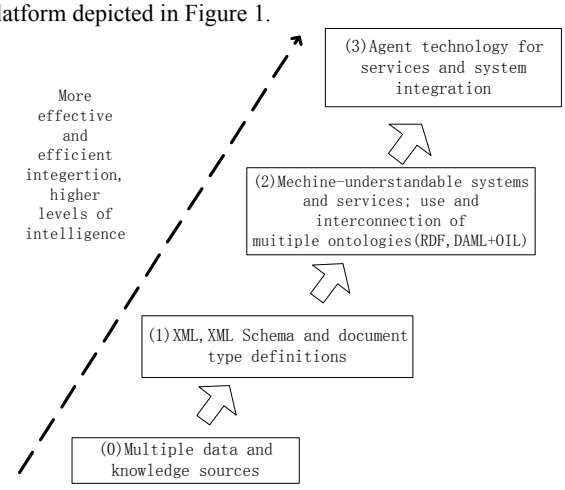


Figure 1 The steps needed to achieve intelligent data and system integration in the machine-understandable way

3. PROBLEMS OF DISTRIBUTED INTEGRATION

3.1 XML data and metadata mining

XML has become an important choice for distributed data representation and exchange. A number of XML-based 'standards' have been proposed in different domains, including Synchronized Multimedia Integration Language (SMIL)(http://www.w3.org/TR/REC-smil), Mathematical Markup Language (MathML) (http://www.w3.org/TR/REC-MathML), Scalable Vector Graphics (SVG) (http://www.w3.org/TR/WD-SVG), Drawing Meta Language (DrawML) (http://www.w3.org/TR/NOTE-drawml), Electronic Business with XML(ebXML)(http://www.ebxml.org/), VoiceXML(http://www.voicexml.org/), and Biopolymer Markup Language (BIOML) (http://www.bioml.com/), etc. One of the most important challenges in the XML standards will be to identifiy novel and useful patterns from large document collections. These XML databases significantly differ from traditional data representation systems [5]. Although several data mining and artificial intelligence techniques have been successfully applied to different knowledge discovery domains, new solutions will be required to approach this problem.

One fundamental point is how to measure similarity between XML-based data or metadata in order to perform search, retrieval and other recognition tasks. These models should approach similarity at different levels, such as metadata, documents, elements and attributes. This factor is linked to the problem of defining operators for element and attribute comparisons, indexing, and the processing of external references. The data mining research community needs to provide users with inexpensive tools to generate, parse and classify XML resources. Moreover, key data mining functions, such as clustering [6], might require new methods due to the multi-layered, structured and heterogeneous nature of XML. Data engineers should also propose XML-based methods to represent analysis results. It may allow, for example, multiple data mining systems to share resources and functions, which might represent an effective way to support integration in distributed application environments.

3.2 Unsupervised concept discovery

Effective data and knowledge integration may be achieved only if there is a good understanding of the embedded semantics of the domain under consideration. At the same time it has been suggested that, in spite the heterogeneity of information systems, it should be possible to identify a semantic unity within a common application domain. This unity may be expressed, for instance, in the form of similar data structures or usage patterns. It may be represented at various levels such as the database model, data structures and in the knowledge background applied by the community of users. Therefore, one of the major objectives for computers is how to support the discovery of this knowledge.

It has been shown, for instance, that a conceptual integration approach may be based on the discovery of similarities at the metadata level [7]. Such a metadata mining process may be performed on database objects or elements to discover sets of classes. Thus it can provide the basis for a conceptual description of the application domain. The classes or concepts discovered might not only represent fundamental components of the domain ontology, but also they may simplify retrieval and navigation tasks. Figure 2 summarizes this concept discovery process based on a metadata clustering algorithm.

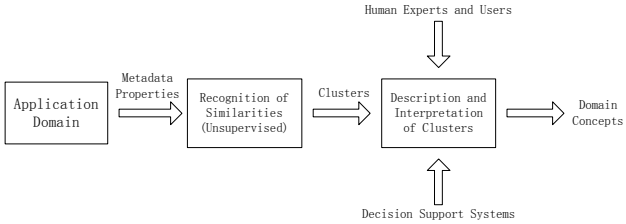


Figure 2 A Concept Discovery Framework to Support Knowledge Integration Tasks

3.3 Service integration

One of the most time-consuming research tasks is to find the right tools or databases from the many available on the web. Thus, one useful application is to automate the processes of finding and executing the services. An intelligent software agent can be defined as a computational program that is proactive, autonomous and able to adapt to new situations [7]. It has been suggested that the implementation of agent communities may facilitate Internet-based knowledge discovery tasks [8]. But the success of this will depend on how we exploit the benefits obtained from using web ontologies.

The semantic markup of web-based systems will represent a knowledge base that will allow agents to exploit users' constraints and preferences for the automatic discovery, execution and composition of web services [9].

- Automatic service discovery. It involves the automatic localization of web tools or applications, based on the properties and preferences specified by a user.
- Automatic web service execution. It consists of automatically executing a web-based application.
- Automatic composition of web services. In this task agents should be able to perform complex functions based on the automatic selection and execution of services. These agents should also generate responses to those processes.

However, none of these applications are entirely available today due to the limitations of the existing web. For instance there is the need to improve the capabilities of markup languages.

DAML [10] and OIL [11] are two important attempts (AI-inspired models) to build on XML, RDF [12] and RDFS [13], and both support for variables, quantification, rules to be well suited for building the Semantic Web. In December 2000, DAML and OIL came into being DAML+OIL [14,15], which has been submitted to W3C. It focuses on the development of declarative representations of web services, domain constraints and user preferences.

This type of project provides the basis for representing the data and metadata associated with a web-based service (including constraints and capabilities), the protocols for its execution and the consequences of it use. These factors are crucial to support the development of agent technologies for service and system integration.

DAML-S (http://www.daml.org/services/) is a DAML+OIL ontology for describing the properties and capabilities of Web Services. Web Services–Web-accessible programs and devices – are garnering a great deal of interest from industry, and standards are emerging for low-level descriptions of Web Services. DAML-S complements this effort by providing Web Service descriptions at the application layer, describing what a service can do, and not just how it does it.

4. Layers of Distributed Data and System Integration Framework

Now, we give the layers of distributed data and system integration framework:

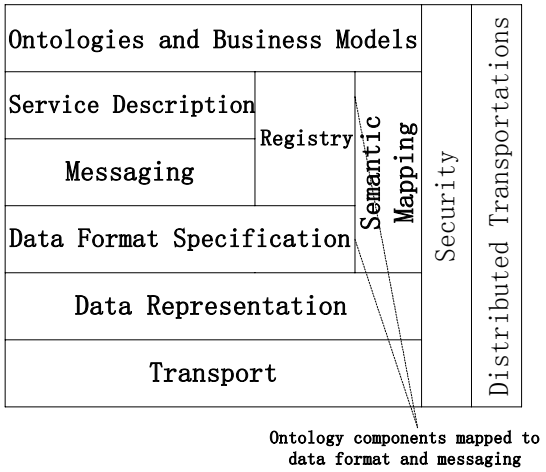


Figure 3 Layers of Distributed Data and System Integration Framework

Each layer is supported by the standards/techniques presented as follow:

Layers	Standards/Techniques
Ontologies and	DAML+OIL
Service Description	WSDL, DAML-S
Messaging Registry	ebXML, SOAP
Data Format	XML DTD, XML Schema,
Data Representation	XML, XML Namespaces
Transport	TCP/IP, HTTP, and FTP

5. CONCLUSIONS

This paper has discussed current research, opportunities and challenges to achieve knowledge integration in the Semantic Web era. Artificial intelligence techniques together with advances in ontologies and semantic markup languages represent a promising approach to simplifying time-consuming research tasks in distributed information environments. Furthermore, this synergy may significantly support knowledge discovery processes in all kinds of fields.

6. REFERENCES

[1] Furukawa K, Michie D, Muggleton S. 1999. Intelligent agents. Oxford University Press: Oxford; 515.
[2] Fensel D. 2000. The semantic web and its languages. IEEE Intelligent Systems 15: 67–73.
[3] McIlraith S, Son T, Zeng H. 2001. Semantic web services. IEEE Intelligent Systems 16: 46–53.
[4] Hendler J. 2001. Agents and the semantic web. IEEE Intelligent Systems 16: 30–37.
[5] Bertino E, Catani B. 2001. Integrating XML and databases. IEEE Internet Computing 5: 84–88.
[6] Soumen Chakrabarti. 2000. Data mining for hypertext: A tutorial survey. SIGKDD Explorations, ACM SIGKDD, Jan 2000. Volume 1, Issue 2.
[7] Srinivasan U, Ngu A, Gedeon T. 2000. J Am Soc Inf Sci 51:707–723.
[8] Hendler J, McGuiness D. 2000. The DARPA agent markup language. IEEE Intelligent Systems 15: 72–73.
[9] McIlraith S, Son T, Zeng H. 2001. Semantic web services. IEEE Intelligent Systems 16: 46–53.
[10] http://www.daml.org/
[11] D. Fensel et al., “OIL in a nutshell”, Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000), R. Dieng et al. (eds.), “Lecture Notes in Artificial Intelligence”, LNAI, Springer-Verlag, October 2000. M.C.A. Klein et al. “The Relation between Ontologies and Schema-Languages: Translating OIL Specifications to XML-Schema”, Proceedings of the Workshop on Applications of Ontologies and Problem-solving Methods, 14th European Conference on Artificial Intelligence ECAI-00, Berlin, Germany August 20-25, 2000. http://www.ontoknowledge.org/oil/
[12] Ora Lassila, Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification, W3C Recommendation 22 February 1999, http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/.
[13] D. Brickley, R. Guha (eds.): Resource Description Framework (RDF) Schema Specification, W3C Candidate Recommendation 30 April 2002, http://www.w3.org/TR/2002/WD-rdf-schema-20020430/.
[14] Dan Connolly, Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein. Annotated DAML+OIL Ontology Markup, W3C Note 18 December 2001, http://www.w3.org/TR/2001/NOTE-daml+oil-walkthru-20011218
[15] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description, W3C Note 18 December 2001, http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218

Cost-based Proxy Caching

Rassul AYANI
Royal Institute of Technology
Department of Microelectronics and Information Technology
164-40, Kista, Stockholm, SWEDEN
E-mail: rassul@it.kth.se
And
Yong Meng TEO and Peng CHEN
Department of Computer Science
National University of Singapore
3 Science Drive 2
SINGAPORE 117543
E-mail: teoym@comp.nus.edu.sg

ABSTRACT

Caching has been used for decades as an effective performance enhancing technique in computer systems, among others. However, the traditional cache replacement algorithms are not easily applicable to WWW applications. Moreover, the frequently used hit-ratio and byte-hit ratio are not appropriate measures in WWW applications, because non-uniformity of the object sizes and non-uniformity cost of cache misses in WWW traffic. In this paper, we propose a cost efficient cache replacement algorithm, CERA. We define Cost Reduction Ratio (CRR), as the cost saved by using a cache divided by the total cost if no cache was used. We compare performance of CERA with other caching algorithms using CRR, hit-ratio and byte-hit ratio as performance metrics. Our experimental results indicate that CERA outperforms LRU, LFU and GDS.

1. INTRODUCTION

Caching is an effective performance enhancing technique that has been used in computer systems for decades. However, proxy caching differs substantially from the traditional ones. Two main features of the World Wide Web applications that differ from the traditional caching are (i) non-uniformity of the object sizes and (ii) non-uniformity of the cost of cache misses (as opposed to the traditional caching where all cache blocks have the same size and require the same amount of time to be retrieved in case of cache misses). The tradition metrics for measuring caching efficiency has been hit-ratio (HR), which is defined as the number of requests satisfied by the cache divided by the total number of requests. Obviously, HR is not an appropriate metric to measure performance of proxy caches, because of the non-uniformity of object sizes and non-uniformity cost of misses. Some researchers have suggested byte hit ratio (BHR), which is defined as the number of bytes found in the cache divided by the total number of bytes requested within the observation period. The BHR takes into consideration the non-uniformity of the object sizes, but it fails to consider the non-uniformity cost of misses. Similar to the Delay Saving Ratio (DSR), which was suggested by Shim [[18]], we introduce *Cost Reduction Ratio* (CRR) to measure caching efficiency in reducing the overall cost of data retrieval. We propose a cost efficient proxy caching algorithm (CERA) and compare its performance with several of the other caching algorithms in the literature, including the Greedy Dual-Size (GDS) described in [[4]].

The rest of the paper is organized as following: In section 2 we discuss some metrics for measuring performance of Proxy caches are discussed and in Section 3 we propose a cost efficient caching algorithm. In section 4 we compare performance of our algorithm with LRU, LFU and GDS. Some conclusions are given in section 5.

2. COST BASED CACHING

Generally, the users of Internet can be grouped as: (i) Clients who are looking for a shorter response time, and (ii) Clients who seek to maximize utilization of the bandwidth (for instance an Internet Service Provider, ISP). For the former group the average response time should be minimized, whereas in the latter case the throughput must be maximized. Thus, we employ two cost models to optimize proxy caching for the two targeted user groups. Firstly, a latency model that would measure the download latency perceived by the end users, and secondly a traffic model that would measure the generated network traffic.

A cache miss incurs a cost for fetching the missing object from its source server. The cost can be measured in two different ways, namely by: (i) the amount of traffic generated to retrieve the missing objects, and (ii) the download latency experienced by the end users to retrieve the missing objects. We define the cost reduction ratio (CRR) in Equation 1.

$$\textbf{CRR} = \frac{\sum H_i * C_i}{\sum C_i} * 100\%, \text{ where } H_i = \begin{cases} 1, & \textit{if request I is a HIT} \\ 0, & \textit{Otherwise} \end{cases}$$

where C_i is the cost for retrieving object i using the "traffic" or the "download latency" definition.

Equation 1 Definition of Cost Reduction Ratio

It should be noted that HR, BHR and CRR measure different aspects of using a proxy cache in a network. A high HR demonstrates that fewer requests being retrieved from the original servers (i.e., more clients experienced shorter response time). However, a high HR does not mean necessarily a lower cost because of the non-uniformity of objects sizes and retrieval costs. A high BHR indicates that fewer bytes have been retrieved from the sources and thus less traffic has been generated, but it doesn't mean necessarily a lower cost because

of the non-uniform cost of retrieval. But CRR measures the cost saving as a result of caching; higher CRR means lower cost (in term of network traffic or perceived latency, depending on which definition of cost being applied).

3. CERA: A COST EFFECTIVE REPLACEMENT ALGORITHM

The LRU is a well performing cache replacement algorithm for applications exhibiting good locality with uniform object size and uniform cost of misses [[3]]. However, in the web context, the non-uniformity of object sizes and the non-uniform cost of misses make the LRU algorithm less attractive.

We have developed a Cost Effective Replacement Algorithm (CERA) to reduce the overall retrieval cost. We define cost as the download latency perceived by the end user in our latency model, and the amount of generated network traffic in our traffic model. In this paper, CERA(L) and CERA(P) denote CERA using latency model and packet model respectively.

In CERA, a benefit value (BV) is assigned to each object representing its importance in the cache. When the cache is full, the object with the lowest BV is replaced. Unlike caching in operating systems, web requests refer to objects with varying size and varying retrieval cost of retrieval cost. Hence, CERA considers miss cost, object (or file) size, and access frequency. As shown in equation 2, the BV consists of three parts: normalized cost, re-accessing probability (denoted by P_r) and dynamic aging.

3.1 Re-accessing Probability

The normalized cost of an object is the download latency per byte for the latency model and the amount of communication per byte in the packet model. The re-accessing probability, P_r, indicates the object's future popularity. The computation of P_r is rather complicated and will be discussed in the full paper. We consider frequency, size and Zipf's law to estimate P_r, where frequency is defined as the number of previous accesses to an object.

If an objects has been accessed f times up to now, we estimate the probability of being re-accessed again in the future, P_f, as D_{f+1}/D_f, where D_f is the number of documents accessed at least f time. In our simulation, a background process collects the number of accesses to each object and then the new P_f values are calculated periodically. Although P_f values are not accurate, they should represent good estimates for P_r in the steady state.

3.2 Impact of Size on Re-accessing Probability

It has been shown that Web requests have a strong preference for smaller files [[6], [7], [8], [13], [15]]. Furthermore, it has been advocated that the average access rate of an object can be estimated given its size. Let R be the average access rate to an object and S is its size. R can be estimated as $\mathbf{R=C/S^b}$, where C and b are two constant factors [[18], [19]]. However, it is well known small files often dominate that web workload. Consequently, using the size of the object directly to estimate its re-accessing probability may over-compensate the small files. To avoid such situations, a logarithmic scale is often used to estimate the average access rate [[1]]. Hence, we estimate the re-accessing probability considering object's size in Equation 3.

$$BV = (\text{Cost} / \text{Size}) * Pr + \text{Age}$$

Cost Cost of retrieving the object from the original server.
It is measured as the download latency or the amount of generated traffic:
- Latency model: `SERVER_DURATION` field in DEC trace
- Packet model: Cost = 2 + Size / 536, SJ trace

Pr Probability of re-accessing an object

$$\boldsymbol{Pr} = \left(\frac{P_f}{(\text{Log}_{10}\text{Size})^b} \right)^{1/\alpha}, \text{ where } P_f = D_{f+1} / D_f$$

- P_f: Conditional probability of re-accessing an object that has been accessed f times
- D_f: Number of documents which have been accessed at least f times
- α: Characteristic value of Zipf's-like distribution; set to 0.77 for DEC and SJ
- b: A constant that weights the effect of *Size* factor; set to 1.3 for DEC and SJ

Size Size of the requested object

Age Age of the cache defined as the minimum BV of all objects in the cache

Equation 2 The Complete Expression of Benefit Value for CERA

$$P_s = \frac{C}{(\text{Log}_{10}\text{Size})^b}, \text{ where b = 1.3 and C is a constant}$$

Equation 3 Re-accessing Probability Considering Size

$P = \frac{K}{R^\alpha}$, where R is the page's popularity ranking, and K and α are two workload dependent parameters

Equation 4 A General Model representing a Zipf's-like Behavior

It has been shown that web requests do not follow exactly Zipf's law. Brelau introduces a model for web requests, shown in equation 4, which follows a Zipf's-like behavior [[6], [7]].

3.3 Aging Policy

In CERA, we use a dynamic aging policy similar to the one proposed by Arlitt and Dilley [[5]]. In our aging scheme, the cache age is the time of the latest access. When an object is brought to the cache its BV is set to its retrieval cost plus H (initially H=0). In case of cache hit, H is set to the current time and the BV is set equal to its retrieval cost + H. In this way, whenever an object is accessed its BV is updated (its BV is increased), but the BV of those objects that are not accessed remains unchanged.

4. PERFORMANCE MEASUREMENT

In this section, we compare the performance of CERA with LRU, LFU [[16]] and the GDS (Greedy Dual Size) [[4]] using a trace driven simulator. We selected three traces collected by different sources to test the performance of proxy cache replacement algorithms. Since small and large traces exhibit different behavior, we chose a short 3-day trace and two longer 30-day traces. The 3-day trace (referred to as **DEC**) was collected by Digital Equipment Corp [[10]], the other two are

downloaded from the National Laboratory for Applied Network Research, NLANR [[10]], and are referred to as **SJ** and **BO** respectively. These traces represent different types of workload. The proxy server for DEC trace is a corporate proxy with heavy web traffic servicing over 10,000 clients. The original DEC trace contained four weeks of data, but we extracted three days of the requests. Unlike DEC trace, SJ trace was collected for a regional proxy server at MAE-West Exchange Point in San Jose, California [[10]].
Table **1** illustrates major characteristics of DEC, SJ and BO traces.

Table 1 Major Characteristics of the Three Traces

Property	DEC Trace	SJ Trace	BO Trace
Trace size	262.4 MB	292.4 MB	658.5 MB
Duration	3 days	30 days	30 days
No. of all requests	2,133,953	2,163,985	4,716,277
No. of distinct requests	857,914	960,818	2,480,375
Size of all requests	21.1 GB	41.6 GB	81.5 GB
Size of distinct requests	11.1 GB	19.3 GB	37.8 GB
Average size of distinct requests	12,880 Byte	20,071 Byte	15,241 Byte
Standard deviation of distinct requests	99,551 Byte	215,887 Byte	601,202 Byte

We have used three traces, but only the results of two of them are presented in this report. The results of DEC trace are shown in Figures 1 to 3 and the results of BO trace in Figures 4 to 6.
In our performance evaluation, CERA(L) and CERA(P) denote CERA for latency and packet cost model respectively, and similarly GDS(L) and GDS(P) denote the GDS algorithm for latency and packet cost model respectively.

4.1 DEC Trace

Figure 1 compares the cache performance for cost based algorithms (CERA and GDS), LRU and LFU in term of CRR. CERA(L) outperforms the other replacement algorithms in term of CRR for all cache sizes. The CRR obtained by CERA is up to 140% better than LFU and LRU for a 10MB cache. While in a 1GB cache CERA(L) performs 23 % better than LFU. GDS, which considers cost, size and aging policy, performs much better than LRU and LFU. But, CERA(L) performs better than GDS(L) and its relative improvement ranges from 24% for small cache to 5% for the largest cache size.
CERA(P) is also better at reducing cost than LRU, LFU and GDS. More concretely, performance of CERA(P) is 67% higher than LRU for a 50MB cache and 50% higher for 100 MB cache .
Figure 2 compares performance of the algorithms in term of HR. With moderate cache sizes (200MB and 400MB), CERA(P) gives about 50% HR improvement compared to LRU and LFU. The HR of GDS(P) is a little better than LRU and LFU. GDS(P) does not consider cold cache pollution, therefore its performance is not well optimized.
Figure 3 compares performance of the algorithms in term of BHR. CERA(P) is about 10% better than LRU and 20%-30% better than LFU. However, CERA(L) does not illustrate any impressing performance in term of BHR.

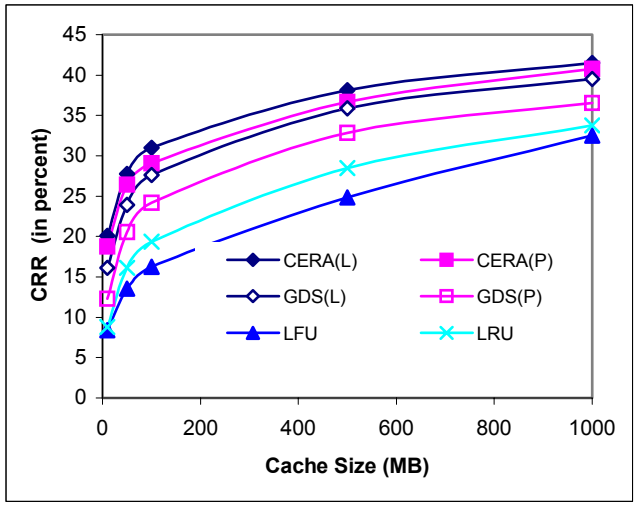


Figure 1 DEC Trace - Cost Reduction Ratio

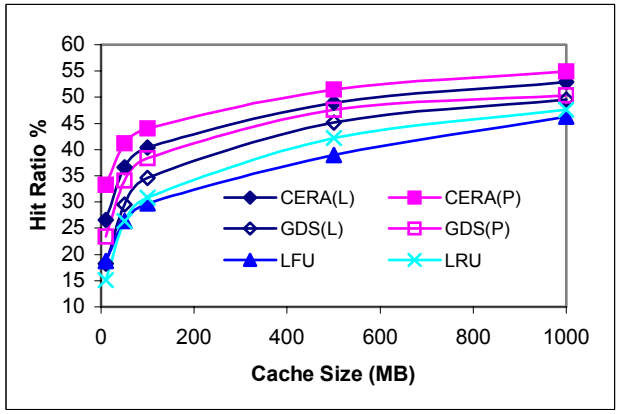


Figure 2 DEC Trace - Hit Ratio

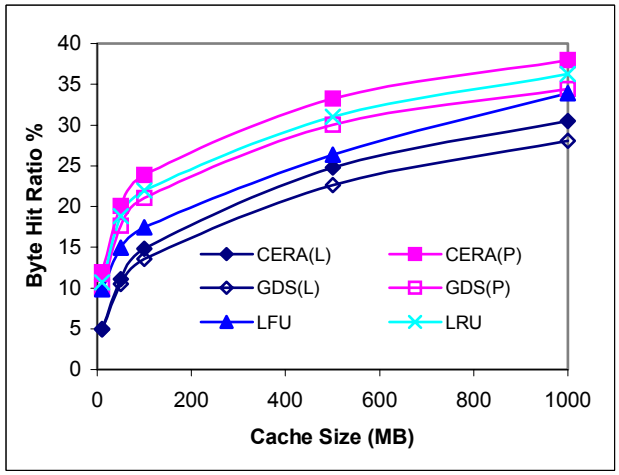


Figure 3 DEC Trace - Byte Hit Ratio

4.2 SJ Trace

Since the download latency for each request is not included in this trace [[10]], we are not able to calculate the BV for CERA(L). Hence, for this trace we measure performance of CERA(P) and compare it with GDS(P) , LRU and LFU. The performance results are shown in Figure 4, Figure 5 and Figure 6.

Similar to the DEC trace, CERA(P) consistently performs best in reducing the cost of retrieval (see Figure 4). However, the

degree of improvement is not as high as for the DEC trace. When the cache size is small (18MB), CERA(P) outperforms GDS(P) by around 5% and exceeds LFU by around 13%. In terms of HR (Figure 5) and BHR (Figure 6), CERA(P) still outperforms the other replacement algorithms as it did for DEC trace.

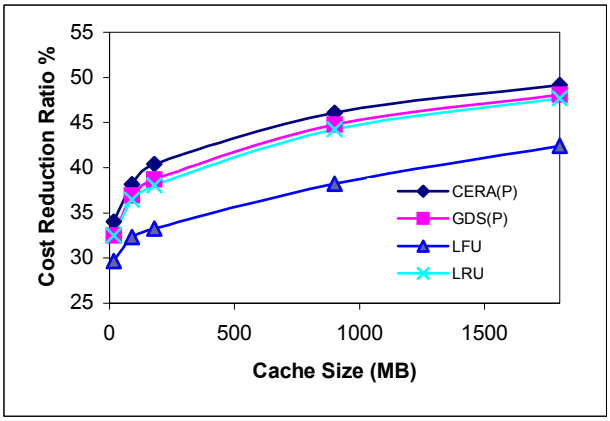


Figure 4 SJ Trace - Cost Reduction Ratio

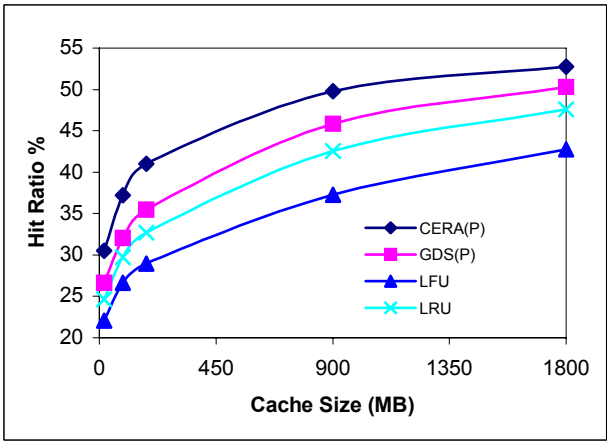


Figure 5 SJ Trace - Hit Ratio

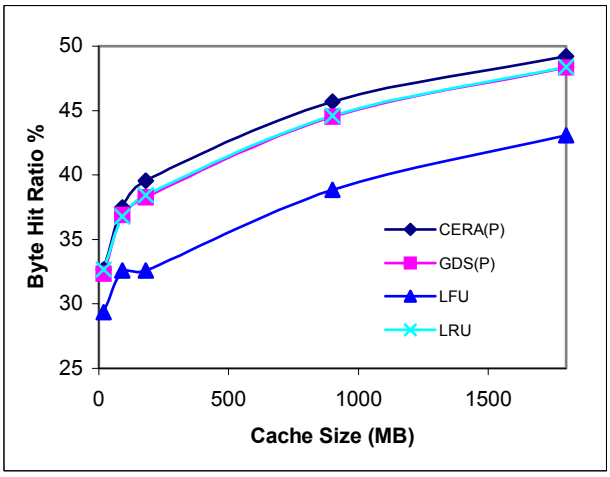


Figure 6 SJ Trace - Byte Hit Ratio

Note that there are two main differences between the SJ and DEC results. First, GDS(P) does not show significant improvements in reducing the cost for SJ Trace (Figure 4). Second, CERA(P) has the best BHR but the performance improvement is small compared to the improvement in DEC trace (compare Figure 3 and Figure 6).

In CERA, one important component of the BV is the predicted re-accessing probability. With CERA, objects that are predicted be accessed again in the future stay in the cache for a longer time. Consequently, if the workload demonstrates a high short-term temporal locality, there will be more hits in a cache with CERA.

4.3 BO Trace

The simulation results for the BO trace are similar to those for the SJ trace because they represent similar workload characteristics. In term of CRR, CERA(P) is the best of the four algorithms (see Figure 7). When the cache size is set to 5% and 10% of the total size of the distinct objects, CERA(P) is only 3 - 4% less than its upper bound (i.e.., the 54.3% improvement achieved for infinite cache size).

Similar to DEC and SJ trace, CERA(P) performs best in term of CCR and HR. The superiority is even more significant than the previous traces. With 180MB cache size, CERA(P) has a HR that is 70% higher than LRU, 35% higher than LRU and 24% higher than GDS(P). The improvements are remarkable for other cache sizes too (see Figure 8). However, CERA(P) is not always the best in term of BHR for BO trace (see Figure 9) . For larger cache sizes it has the best BHR, but not for small caches. Considering the performance improvements of CERA in term of CRR and HR, the slight decrease in BHR for smaller cache sizes is negligible.

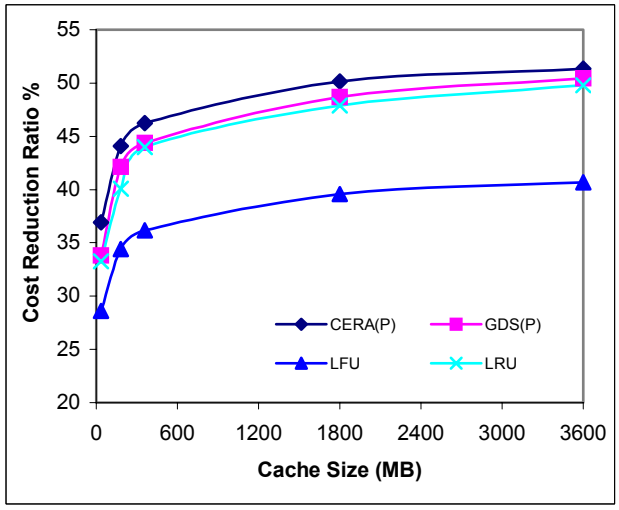


Figure 7 BO Trace - Cost Reduction Ratio

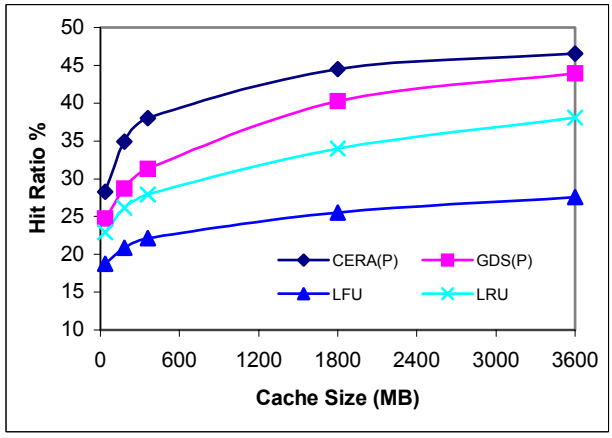


Figure 8 BO Trace - Hit Ratio

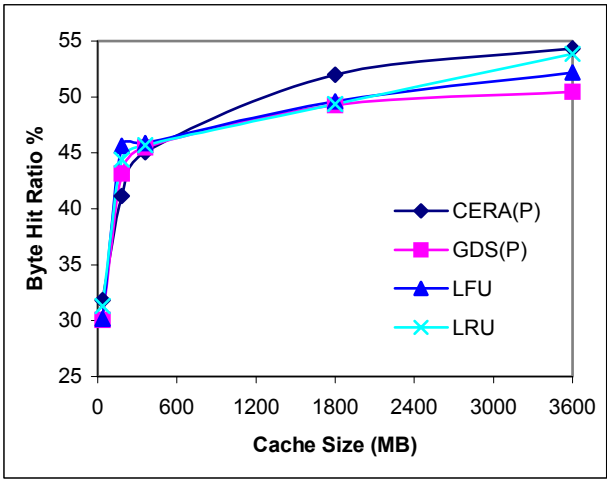


Figure 9 BO Trace - Byte Hit Ratio

5. CONCLUSIONS

We proposed a proxy cache replacement algorithm CERA, which consists of three parts: *cost*, *re-accessing probability* and *aging*. We conducted several experiments using three traces and compared performance of CERA with LRU, LFU and GDS. As metrics we used hit-ratio (HR), byt-hit-ratio (BHR) and cost reduction rate (CRR) to measure and compare performance of these algorithms. Our experimental results show that:

- CERA with the latency model, denoted as CERA(L) consistently outperforms the other algorithms in term of CRR.
- CERA with the packet model, denoted as CERA(P) consistently outperforms the other algorithms in term of HR.
- CERA(P) outperforms the other algorithms in term of BHR for large caches, but it shows slightly lower performance for small caches.
- The performance improvement of CERA over the other algorithms depends on the workload characteristics, but it performs best for workloads exhibiting good short-term temporal locality.

6. ACKNOWLEDGEMENTS

This project is supported by a joint research grant from Fujitsu Computers (Singapore) Pte Ltd and the National University of Singapore.

7. REFERENCES

[1] A. Iuutonen, "*Web Proxy Servers*", Prentice Hall, December 1997.

[2] P. Lorenzetti and L. Rizzo, "*Replacement Policies for A Proxy Cache*", http://www.iet.unipi.it/~luigi/caching.ps.gz, 1998.

[3] P. Cao and S. Irani, "*Cost-Aware Proxy Caching Algorithm*", Proceedings of the 1997 USENIX Symposium on Internet Technology and Systems, December 1997.

[4] A. Bestavros and S. Jin, "*Greedy Dual* Web Caching Algorithm, Exploiting the Two Sources Of Temporal Locality in Web Request Streams*", 5th International Web Caching and Content Delivery Workshop, Lisbon, Portugal, 22-24 May 2000.

[5] M. Arlitt and J. Dilley, "*Improving Proxy Cache Performance: Analysis of Three Replacement Policies*", IEEE Internet Computing, November/December 1999.

[6] L. Brelau, P. Cao, F. Li, G. Phillips and S. Shenker, "*Web Caching and Zipf-Like Distributions: Evidence and Implication*", IEEE Infocom, Vol. XX, No. Y, Month 1999.

[7] L. Brelau, P. Cao, F. Li, G. Phillips and S. Shenker, "*On the Implications of Zipf's Law for Web Caching*", http://www.cs.wisc.edu/~cao/papers/zipf-implication/.

[8] M. Arlitt and C. Williamson, "Web Server Workload Characterization: the Search for Invariant", SIGMETRICS 96, May 1996, PA, USA

[9] M. Abrams and R. Wooster, "*Proxy Caching that estimates page load delays*", Technical Paper 24061-0106, Virginia tech Network Research Group, Computer Science Dept, Blacksburg, VA, December 1999.

[10] Digital Equipment Cooperation, Digital Web Proxy Traces, ftp://ftp.digital.com/pub/DEC/traces/proxy/webtraces.html.

[11] National Laboratory for Applied Network Research, Squid 2.0 Web Proxy Traces, ftp://ircache.nlanr.net/Traces/.

[12] P. Barford, A. Bestavros, A. Bradely and M. Vrovella, "*Changeds in Web Client Access Patterns, Characteristics and Caching Implications*", BUCS-TR-1998-023, November 1998.

[13] A. Mahanti, "*Web Proxy Workload Characterization and Modeling*", A thesis submitted in Dept. of Comp., Univ. of Saskatchewan, September 1999.

[14] E.J. O'Neil, P.E. O'Neil, G. Weikum, "*The LRU-K page replacement algorithm for database disk buffering*", Proceedings of ACM SIGMOD, 1993.

[15] A. Bestavros, M. Crovella and C. Cunha, "*Characteristics of WWW Client-Based Traces*", Technical Report TR-95-010, Boston University, Apr. 1995.

[16] Z. Liu,, N. Niclausse, and P. Nain, "*A New Efficient Caching Policy for WWW*", Proceedings of the 1998 Internet Server Performance Workshop (WISP '98), Madison, WI, pp.119-128, June 1998.

[17] R. Karedla, J.S. Love, B.G. Wherry, "*Caching Strategies to Improve Disk System Performance",* IEEE Computer, pp. 38-46, Vol. 27, No. 3, March 1994.

[18] J. Shim, P. Scheumermann and R. Vingralek, "*Proxy Cache Algorithms: Design, Implementation, and Performance*", IEEE Transactions on Knowledge and Data Engineering, Vol. 11, No. 4, July/August 1999.

[19] C. Cunha, A. Bestavros and M. Crovella, "*Characteristics of WWW Client-Based Traces*", Technical Report TR-95-010, Boston Univ. Apr. 1995.

[20] G. K. Zipf. "*Relative Frequency as A Determinant of Phonetic Change*", Reprinted form the Harvard Studies in Classical Philology, Vol XL, 1992.

Design and Research on a Novel Fair Exchange Protocol Applied to Electronic Trading

Liu Quan Wang Shen ZhouZude
School of Information Engineering, Wuhan University of Technology
Wuhan 430070, Hubei, China
E-mail: qliu@public.wh.hb.cn

ABSTRACT

With the development of Internet technology and the rapid increment of Intranet application, the electronic commerce application is becoming a hotspot. But in the environment of Internet-based E-commerce, there are many unfair exchanges. So in this paper, a new fair exchange protocol with off-line semi-trusted third party is proposed. It is efficient, and all messages are encrypted naturally. It is based on publicly verifiable secret sharing (PVSS) theory.
Keywords: Electronic trading, fair exchange protocol, publicly verifiable secret sharing, Internet security

1. INTRODUCTION

The fair exchange protocol is used to assure the exchange's impartial process. And what the impartiality here refer to is that if the both sides of the deal are honest, then at the end of the dealing, they both get what they want from the other and succeed in the dealing. on the other hand, if one of the two are honest such as withdrawing prematurely or acting dishonorably, the other will not be cheated and have a loss. Impartiality is the principal demand of the exchange protocol.

The fair dealing is an old issue, when the human began the exchange, the search of the principle of the fair exchange also began. And recently, with the fast development of the computer internet, especially the repaid popularity of the internet, how to make the fair exchange on the internet on the line has become the hot topic, and aroused the interest and attention of the academician and governments all over the world.

The use of the creditable third party is an obvious way to solve the problem in the fair exchange. That is, the two parties A and B, give their respective secrets S_A and S_B to the third party. Then the third party validates the authenticity, if they are authentic, the third party will give S_A to B and gives S_B to A. however the utterly depend on the third party is not practical in most cases. One is that from the point of the security, the third party uncertainly exists(even the third party is willing to , it will not necessarily do under the attach of the hack) and on the other hand ,from the point of capability, it is also easy to be the choke point of the system.

In order to find a practical way, broad researches have been made. And generally, it can be divided into two kinds. one kind is not to use the third party. The principle is the both let out the secrets to the opposite by degrees. for example, the secrets can be divided into n shares, and in one round, one of them is exchanged. If the exchange is terminated after m (m<n)rounds, (for one 's intent or the intermit of the communication way),the both will try to guess the remained secrets of the other's. if their ability to speculate is not equivalent (such as the ability of calculation , the possess of resource and so on) ,the impartiality of the exchange will be broken. But it is unpractical to require the equal ability of the both in the real system, especially on internet. Furthermore, it requires a lot of communication, and is of low efficiency.

The other way is to use the third party. Different from the former, in this way, the exchange of the secret depends on the third party, and is completed at a time. So the communication needed is much less than the former .this kind of protocol differs with the role of the third party in the exchange, the volume of the information, the status of the dispatcher and the receiver and the requirement of the substrate communication lines. Enlighten under the Asokan protocol and Franklin protocol and the based on publicly verifiable secret sharing theory, The author puts forward a new online exchange protocol, which makes use of the advantage of the two and betters Their shortcomings. In the practice of the e-commercial business, this paper is very valuable.

2. THE COMMENT ABOUT SOME CORRELATIVE PROTOCOLS

Asokan Protocol

Asokan puts forward a optimization protocol for concerning net fair exchange digital signature. It is the same with any familiar digital signature scheme. We take example with the Schnorr scheme for the simple discussion.[1]

Suppose $G= Z_p^*$, among them, the p is a big prime number, p-1 to reach a big prime number factor the q.g is a Z_p^* is q subgroup born dollar for G_q. The term choose the $x \in_R Z_q$, calculate $h= g^x \bmod p$. In Schnorr scheme, x is the private key, p,q,g,h make up of the public key. When someone is in the signature, will choose $r \in_R Z_q$, calculate z = cx+r, among them $c= H(g^r, m) \in Z_q$, H is a Hash function. signature is in (c, z). When someone verifies, will check $c= H(g^z h^{-c}, m)$.

When A want to get B's signature about the information m_B, B want to get A's signature about the information m_A, Asokan's protocol will work like the following ways:

Step 1: A calculates m_A's Schnorr signature (c_A,z_A) and it's reduced signature (c_A,u_A). Among them, $u_A = g^{z_A} \bmod p \in G_q$. A sends (c_A,u_A) to B.

Step 2: B validates $c_A = H(u_A h_A^{-cA},m_A)$. If it is invalidation, B will stop. Otherwise, B calculates m_B's Schnorr signature (c_B,z_B) and it's reduced signature (c_B,u_B), among them, $u_B = g^{z_B} \bmod p \in G_q$. B sends (c_B,u_B) to A.

Step 3: A validates $c_B=H(u_B,h_B^{-cB},m_B)$. If it is invalidation, A will stop. Otherwise, A encrypts z_A to get α by the third part public key, and provides evidence $PROOF_A$, to prove for B zero knowledge, the third part will get $log_g u_A$ when they decrypt α.

Step 4: If B can't accept $PROOF_A$, B will stop. Otherwise, B encrypts z_B get β by the third part public key, and provides evidence $PROOF_B$, to prove for A zero knowledge, the third part will get $log_g u_B$ when then decrypt β.

Step 5: If A can't accept $PROOF_B$, these show B is deceiving, A will run Abort sub-protocol[2], and inform the third part that don't decrypt α for B. Otherwise, A will send z_A to B.

Step 6: B validates $u_A= g^{zB} \bmod p$. If it is invalidation, B will run B-Resolve sub-protocol[2], resume z_A by the third part. Otherwise, B send z_B to A.

Step 7: A validates $u_B=g^{zB} \bmod p$. If it is invalidation, A will run A-Resolve sub-protocol, resume z_B by the third part.

Step 1~2 can validate the signature validity, it's the special

step for exchanging signature.
Under the mass circumferences, the two exchanging parts are all honest. So, Asokan's protocol only comes down to the two exchanging parts in the normal circumferences (Step3~6). When any part deceives, the fool part need the help from the third part, to take on the beguiler. In this way, the traffic will reduce and the load of the third part will be lighten. The possibility which become the capability bottleneck will debase with them. The third part will work in the out-line circumference, this is the most excellent capability embodiment.
Asokan used the encryption techniques that can be validated, for the third part can help the fool part when the suddenness circumferences happens, he can encrypt each two part's information by the third part's public key (Step3~4). At the same time, the third part will know the everything about the business. It's the protocol's shortage. We will improve on it.

Franklin Protocol

Franklin designs a fair exchange protocol based on the third semi-trusted part. They think the problems over in the finity-group.
First, they define two functions: *f,F*, which have the following properties:
① *f*: $G \to G$, it's a unilateralism non-conflictive function, or is a function that has some normal conflictions which can be disposed easily.
② *F*: $G \times G \to G$, which conform to the requirements that is $F(x,f(y)) = f(xy)$.
We study the basic protocol, some symbols in it have be done some adjustments.
It's the original state of the protocol that one of the exchanging parts *A* own the private key k_A and the unilateralism function $f(k_B)$ of the private key owned by the other part *B*; These is a analogous thing, *B* own the private key k_B and the unilateralism function $f(k_A)$ of the private key owned by the other part *A*. All of them want to exchange the private key with the help from semi-trusted third party (*STTP*) .
We show the process which A send the message m to B by the symbol $A \to B$. The following is the protocol working process:
First, one of the exchanging parties *A* choose a randomly in the circumscription field of *f* and send message a to the other party *B*:

$$A \to B: a,$$

Meanwhile, *B* choose *b* randomly in the circumscription field of f and send message *b* to the other party *A*:

$$B \to A: b,$$

Afterward, *A* will send the following messages to the semi-trusted third party STTP when *A* receives *b*:

$$A \to STTP: f(k_A), f(k_B), k_A a^{-1}, f(b),$$

Meanwhile, *B* will send the following messages to the semi-trusted third party STTP when B receives a:

$$B \to STTP: f(k_B), f(k_A), k_B b^{-1}, f(a),$$

Last, if *STTP* receives the following message from *A*:

$$\alpha_A, \beta_A, \gamma_A, \eta_A,$$

and receives the following message from *B*:

$$\alpha_B, \beta_B, \gamma_B, \eta_B,$$

STTP will validate the two following equations:

$$\alpha_A = \beta_B = F(\gamma_A, \eta_B),$$
$$\alpha_B = \beta_A = F(\gamma_B, \eta_A),$$

if all of the equations come into existence, *STTP* will send the following corresponding messages to *A* and *B*:

$$STTP \to A: \gamma_B . STTP \to B: \gamma_A$$

The basic thinking of Franklin. was derived from the verifiable secret sharing technology[5]. The bargainer (taking *A* for example) share own secret k_A to *B(a)* and the third party (k_{Aa}^{-1}). *B* put own sharing value a by *f(a)*, then the third party can validate it. If the validating process pass, the third party will send own sharing value k_{Aa}^{-1} to *B*, then *B* can resume $k_A (= k_{Aa}^{-1} a)$.
It is the characteristic that unless the both exchanging parties tell the secret the third square actively, otherwise, the third can not know the concrete contents about exchanging(call half can letter the third square).
But, this protocol has 3 shortages. We will improve it on.
1.The traffic is too large, and the protocol is not the most excellent;
2.The third party must work on-line, because it will come down to every running process, it will cause the system's performance bottleneck;
3. And that, if this protocol wants to work on the Internet which have not any physics security, all message must to be encrypt, and can be validated.

3. THE IMPROVED SCHEME

Through our analysis, we can know that the protocols method of Franklin and Asokan seems reach a extremeness: Franklin's method is not the best excellent method, because the third party must work on-line, otherwise the third party can not know the details about the exchanging; Asokan's method is the best excellent method, the third party may work out-line, but he can know the exchanging contents.
We need a fair exchanging protocol, which not only the third party can not know the exchanging contents, but also it's the best excellent method, can work out-line. We will use the public verifiable secret sharing theory.

Public Validate-able Secret Sharing[3],[4]

Shamir brought forward the secret sharing problem firstly in 1979. This model is consist of the secret owner Dealer and numbers n secret sharer $P_1,\ldots,P_n$, as well as a accessing controller structure $A \subseteq 2^{\{1,\ldots,n\}}$. *A* has a monotony quality, which is $Y \in A$ if $X \in A$, $X \subseteq Y$ Y. The secret owner Dealer will run the program Share for sharing the secret s:

$$Share(s) = (s_1, \ldots, s_n)$$

and will send the secret s_i to P_i. The sharer will run the program Recover when he resumes the secret s.

$$\forall X \in A : Recover(\{s_i \mid i \in X\}) = s$$

This secret sharing mechanism (SS) may bring on the cheating from the secret owner and the secret sharer. We brought forward the arithmetic Verify, the sharer can validate the share value use it, which conform to the requirements that is:

$$\exists \mu \forall X \in A : (\forall i \in X : Verify(s_i) = 1)$$
$$\Rightarrow (Recover(\{s_i \mid i \in X\}) = u)$$

If Dealer is honesty, $u = s$.

The Protocol Description

The basic thinking of this scheme is one party of the exchanging two parties *A* will share the own secrets to the other party *B* and the third party *STTP* which is public verifiable, B can validate the share things from A. If the share things is correct, B will share the own secrets to the other party *A* and the third party *STTP* which is public verifiable, *A* can validate the share things from B. If the share things is correct, *A* will send the owned secrets to B and B will send the owned secrets to *A*.
It need the public directory services from standard organization to insure the secrets can't be replaced. We ask the two exchanging parties to save their each messages just like following group to the public directory before the exchanging begins:

$$\langle desc_i, enc_i, f(S_i), sign_i \rangle ,$$

$desc_i$ figures the recommendation to the data file contents; enc_i figures the cryptograph that the date files are encrypted by S_i;

$f(S_i)$ figures the unilateralism function value of the key S_i; $sign_i$ figures the signature of the directory services, to insure we can *desc's* descriptive data file that S_i decrypt enc_i.

Before the exchanging begin, one party of the exchanging *A* own the private key S_A and the unilateralism function value $f(S_B)$ of the other party B's private key; Analogously, B own the private key SB and the unilateralism function value $f(S_A)$ of the other party A's private key. The other thing is each party of *A*,*B*,*STTP* owned the decrypt function D_A,D_B,D_T clandestinely, and the encrypt functions E_A,E_B,E_T is public.

The whole protocol is consist of three sub-protocol:

(1) Normal sub-protocol

Step 1. *A* run the expressions Share(S_A) = (S_{AB},S_{AT}), encrypt the secrets to get S_{AB}、S_{AT} by E_B、E_T, and get the evidences $PROOF_{AB}$ and $PROOF_{AT}$, choose r ($r \in f$) randomly, make a signature to get $SIGN_A$ by the expressions m = (f(r), E_A, E_B, $f(K_A)$, $f(K_B)$), then send it to *B*:

$$mn_1: A \rightarrow B: S_{AB}, S_{AT}, PROOF_{AB}, PROOF_{AT}, SIGN_A.$$

Step 2. *B* run PubVerify to validate the validity of S_{AB}、S_{AT} and $SIGN_A$. If it's wrong, stop it. Otherwise, send the messages to A just like step 1.

$$mn_2: B \rightarrow A: S_{BA}, S_{BT}, PROOF_{BA}, PROOF_{BT}, SIGN_B.$$

Step 3. If *A* is overtime, or *A* run PubVerify to validate the validity of S_{BA}、S_{BT}, and check $SIGN_B$ up, if it's wrong, *A* run to make the sub-protocol abort. Otherwise, send the messages to *B*:

$$mn_3: A \rightarrow B: E_B(S_{AT}).$$

Step 4. If B is overtime, or B check $E_B(S_{AT})$ up, if it's wrong, it run the salvation sub-protocol. Otherwise, it's succeed. *B* send the messages to *A*:

$$mn_4: B \rightarrow A: E_A(S_{BT}).$$

Step 5. If A is overtime, or *A* check $E_A(S_{BT})$ up, if it's wrong, it run the salvation sub-protocol. Otherwise, it's succeed.

(2) Abandoned sub-protocol

Step 1. *A* send the abandoned request to *T*:

$$ma_1: A \rightarrow T: abort, r, SIGN_A.$$

Step 2. *T* checks the request's validity up. If it's invalidation, $ma_2: T \rightarrow A$: "invalidation". Otherwise, if T is in the salvation state, then ma_2: $T \rightarrow A: D_T(S_{BT})$; if T is in the abandoned state, ma_2: $T \rightarrow A$: "abandoned". Send ma_2.

(3) salvation sub-protocol

Step 1.*A/B* sends the salvation request to *T*:

$$mh_1: A/B \rightarrow T: \text{salvation}, S_{AB}, S_{AT}, PROOF_{AB}, PROOF_{AT}, SIGN_A, S_{BA}, S_{BT}, PROOF_{BA}, PROOF_{BT}, SIGN_B.$$

Step 2. T checks the request's validity up. If it's invalidation, $mh_2: T \rightarrow A/B$: "invalidation". Otherwise, if *T* is in the abandoned state, then $mh_2: T \rightarrow A/B$: "abandoned"; if T is in the salvation state, $mh_2: T \rightarrow A/B: D_T(S_{BT})/D_T(S_{AT})$. Send mh_2.

The Analysis about the Protocol

In the following analysis, we suppose the connection between *A* and *B* can be broken forever (it means *A* or *B* abort the exchange); and the connection between A or B and the third party *T* can be resumed within the limited times (the rationality consist the creditability about *T*)

(1) The analysis about the validity

It's obvious that the validity means the exchange will be success if all of three parties abide the protocol and never abort the exchange, and it only comes down to the normal sub-protocol. In this example, it can get *sB* through Recover($D_A(S_{BA})$,$D_A(E_A(s_{BT}))$ with *A*.

(2) The analysis about the equitableness

This point means the two parties in the exchange can get everything they want to get or get nothing after finished the protocol. In this scheme, it need the third party is equitable enough.

(3) The analysis about the effectiveness for a given period of time

This point means the protocol can be finished for a given period of time, and can't break the equitableness in the same time. This protocol is the effectiveness for a given period of time and the time synchronization independent of the network. Any party can decide to finish the protocol according as his local time, and can't break the equitableness.

(4) The analysis about the examine-able

This point means the exchangers can know the error from the third party, whether is intent or accident. This protocol ask the exchangers send the request to the third party, and the third party must give them a answer within the limited time. They can know the conclusion about the third party from the answers.

The improved to Asokan scheme: we know that the third party can get the secret about the exchange when he help the fool. In this scheme, the third party can know nothing except a heft about the secret. The third party can't know any information about the exchange secret because the other heft about the secret is stochastic.

The improving to Franklin scheme: The biggest failing is the third party must be on-line. In this protocol, the third party can work out-line usually and be on-line in the suddenness condition. The traffic will debase. The encrypt and the protocol will be in one part through the PVSS scheme.

4. CONCLUSION

This paper introduced a new fair exchanging scheme based on Asokan and Franklin two fair exchanging schemes, absorbed their excellences, and improved on their shortages. We advanced the research about the fair exchanging protocol in the network. This scheme's characteristic is that the third party can work out-line, and the third party can't know the secret about the exchanging when he helps the fool. This protocol is the high efficiency. With the exchanging practice accumulate in the network, we must face the new request about the protocol, and do more researches to design the perfect exchanging protocol.

5. REFERENCES

[1] Schnorr C P. Efficient signature for smart cards. Journal of Cryptology, 1991, 4(3): 239～252

[2] Asokan N, Shoup V, Waider M. Optimistic fair exchange of digital signatures. In: Advances in Cryptology—Proceedings of EUROCRYPT 98. LNCS1 403, Berlin: Springer-Verlag, 1998. 591～606

[3] Stadler M. Publicly verifiable secret sharing. In: Advances in Cryptology—Proceedings of EUROCRYPT 96. LNCS1070, Berlin: Springer-Verlag, 1996. 191～199

[4] Schoenmakers B. A simple publicly verifiable secret sharing scheme and its application to electronic voting. In: Advances in Cryptology—Proceedings of CRYPT99, LNCS. Berlin: Springer-Verlag. 1999

[5] Chor D, Goldwasser S, Micali S et al. Verifiable secret sharing and achieving simultaneity in the presence of faults. In: Proc of the 26th IEEE Symp on Foundations of Computer Science. Portland, Oregon: IEEE Press, 1985. 383～395

[6] Franklin M K, Reiter M K. Fair exchange with a semi-trusted third party. In: Proc of 4th ACM Conf on Computer and Communication Security. Zurich: ACM Press, 1997.1～5

Methodological Issues for Designing Multi-Agent Systems and Protocols with Authentication and Authorisation for Mobile Environment

Gustavo A. Santana Torrellas
Instituto Nacional de Administración Pública
Secretaría Ejecutiva - Sistema Nacional de Formación Interactiva a Distancia
Km. 14.5 Carretera México-Toluca, Col. Palo Alto, Cuajimalpa
México, D.F., C.P. 05100
Telf: (+5255) 50812681 (dir)
e-mail: userg1514@netscape.net
gustavosantana2002@hotmail.com

ABSTRACT

This paper deals with one of the probably most challenging and, in our opinion, little addressed question that can be found in Distributed Artificial Intelligence today, that of the methodological design of a learning Multi-Agent System (MAS). It relies on three important notions: (1) independence from the implementation techniques; (2) definition of an agent as a set of three different levels of roles; (3) specification of a methodological process that reconciles both the bottom-up and the top-down approaches to the problem of system design. In this paper we show how this method enables services offered over information networks, like Electronic-banking, validating the identity of a user and the authorities he has is a fundamental issue, considering the different levels of behaviours to which they can be applied and the techniques which appear to be best suited in these cases. This presentations allows us to take a broad perspective on the use of all the various techniques developed in Security and their potential use within an MAS design methodology. These techniques are illustrated by examples taken from the Public Key Infrastructure (PKI) based solutions and generally considered to be the most secure and reliable. This allows us to propose in a mobile environment, where the same services can be used through different channels, like the web and the WAP, the issue of authentication and authorisation is often more complex. The standardisation of technologies to be used to solve the problems is advancing at a fast and steady pace. The actual implementations are lagging a step behind, especially when it comes to developing overall solutions for authentication and authorisation.

1. INTRODUCTION

Because of the great interest in using Multi-Agent Systems (MAS) in a wider variety of applications in recent years, agent oriented methodologies and related modelling techniques have become a priority for the development of large-scale agent-based Systems. The work we present here belongs to the disciplines of Telecommunication Engineering, Security, and Distributed Artificial Intelligence. More specifically, we are interested in security-software engineering aspects involved in the development of Multi-agent Systems (MAS). Several methodologies have been proposed for the development of MAS. For the most part, these methodologies remain incomplete: they are either an extension of object-oriented methodologies or an extension of knowledge-based methodologies. In addition, too little effort has gone into the standardisation of MAS methodologies, platforms and environments. It seems obvious, therefore, that software engineering aspects of the development of MAS still remain an open field. The success of the agent paradigm requires systematic methodologies for the specification, analysis and design of MAS applications. We here present a framework of MAS methodologies that enabled us to make an analysis of Authorisation and Authentication Issues in Mobile Environment. Results from this work have also leaded us to propose a unification scheme. Our goal is to make MAS design and development more systematic and to contribute to the standardisation of MAS methodologies and platforms.

2. SOME CONCEPT RELATED MOBILE SECURITY

Authenticating the identity and authorisation of users with a high degree of certainty in open and unreliable environments has been one of the most significant functional problems that have been struggled with, when developing services accessible over the Internet. Many different schemes for accomplishing the desired result have been devised. Some of which have been better than others in terms of user friendliness and the Level of Certainty and Security.
Now that the era of mobile Internet is dawning there is a lot of rethinking of procedures that are used for authentication and authorisation going on. The dilemma that is being faced is that there should exist a mechanism that is as convenient as possible for the users and that would give a perfect certainty of the result of the authentication. For sake of consistency we shall define different terms that we will employ in the subsequent paragraphs.

Authentication
In other words, authentication is the process of determining whether someone or something is, in fact, who or what it claims or is declared to be. There are five methods of authenticating an identity principal [5]:

1. Something the claimant knows
2. Something the claimant owns
3. Something the claimant is
4. Claimant is at a particular place (at a particular time)
5. Authentication is established by a trusted third party.

Authorisation
Authorisation is closely related to authentication. Authorisation is in fact the process of giving someone permission to do or to have something. This implies that in order to be able to give someone access to something this someone needs first to be authenticated. In other words authorisation is logically preceded by authentication. Actual authentication of the identity of an entity or individual is, however, not always required, in order to do the authorisation, if the authorisation can be validated in some other means. Authentication by itself has little use - it is just a means of getting the information needed for authorisation.

Mobile Environment
In the context of this paper the concept 'mobile environment' is used to describe a setting where a user of system accessible through the Internet or other open networks, is not bound to a certain place, device or access channel in order to use the services provided by the systems. An example of a setting like this is a banking system that is accessible through the Internet and WAP access channels.

Public Key Infrastructure and Public-key Cryptography
Public Key Infrastructure (PKI) is "a system for publishing the public-key values used in public-key cryptography" [2]. Public-key cryptography is a technique initially introduced by Diffie and Hellman in 1976 [3]. Public-key cryptography relies on mathematically complex problems dealing with prime number theory, that are impossible to solve in a reasonable time without the correct input - the key. Public-key cryptography is used in essence, like all cryptography used, only for encrypting and decrypting sequences of data. In the public-key cryptography there are fundamentally two keys (a key pair) involved: a key that is kept secret by its owner (the private key) and a key that is made public to everyone (the public key) between which there is no connection that can be discovered. Data that is encrypted using the public key can only be decrypted by using the private key and vice versa. This paper will not discuss the specifics of public-key cryptography and the algorithms related to it in higher Level of detail than this. The assumption is made that the cryptographic algorithms that are being used is strong and the keys that are in use are long enough for the process to be reliable.
Even though, the PKI is not a completely new thing (see [3] from 1976) and it is widely in use almost all over the world, there are some, almost philosophical, debates still going on around it? Generally the fact that the government authorities especially in the United States want to have access to the private keys of all individuals for 'purposes of National security' is being widely debated [1]. Further discussion of these issues is outside of the scope of this paper.
There are actually a number of slightly different approaches taken to implement PKIs but there are at least two things that are common to all of them:
Certification, which is the process of binding a public-key value to an individual or attribute
Validation, which is the process of verifying the validity of the prior.
In his paper "A Survey of Public Key Infrastructures " [2], Branchaud defines certification to be the "means by which public-key values, and information pertaining to those values, are published" and certificates to be "the form in which a PKI communicates public key values or information about public keys, or both". A serious issue to be solved in a PKI and pubic-key cryptography system implementation is the management of the private keys of the individuals. The keys should be kept secret, but the owner of the key should always have access to it.

Certificate Authorities and Digital Certificates
Certificate Authority (CA) is an entity that issues and verifies digital certificates. Digital certificate is a piece of data, containing a public key and its attributes, as the information about its owner and the signature of a CA. The fact that the certificate is signed by a CA, makes it possible for a user of the certificate, e.g. a person who wants to sign something with the public key of the presumed owner of the certificate (i.e. the subscriber of the certificate), to easily gain certainty that the information contained in the certificate is valid.
This is possible since the CAs are trusted entities and the signature of the CA makes also the certificate trusted. In a PKI there normally exists also a Network or hierarchy of CAs [2, 6]. This makes it possible, that as long as a CA is trusted by some CA that is trusted by the root CA (that is implicitly trusted) or a CA on the 'trust path' to the root, that CA is also trusted.
It should be noted that when a CA issues a certificate by digitally signing it, it has to be sure that the public key and the information that is being attached to it in the certificate, i.e. the information about the subscriber of the certificate are correct and coherent. The CA normally relies on a register authority (RA) to do so.

SPKI - Simple Public Key Infrastructure
The SPKI has been defined in IETF RFC 2693 [4]. Idea of SPKI certificates is to link an authority to a key. This differs from the traditional PKI thinking, where identification information is linked to a key in a certificate. In other words the SPKI certificates are more focused on binding rights and authorities to keys than the individuals' identification information to keys, which is the case in with conventional PKI-implementations as the X.509 [4].
For the purposes of open environments this is convenient, since in this setting a degree of anonymity is provided - a user can present a certificate that issues him authority to access some services or resources, without necessarily revealing his identity. The trust is based on a trust Network and delegated authorisation.
Even though it is an ideal method for providing authentication and authorisation mechanisms in open environments, there aren't actually any implementations or products that would support it on the market currently.

X.509 PKI
X.509 is the most widely used PKI standard. It defines all the aspects of the infrastructure, like the structures of certificates and the hierarchical relationships between the CAs.
The X.509 certificates have the following content:

- Version that describes the version of the encoded certificate
- Serial number that is a unique identification number of the CA's signature algorithm identifier
- Issuer name that identifies the CA who signed and issued the certificate
- Validity that indicates period of validity of the certificate
- Subject identifying the holder of the private key
- Subject public key information
- Issuer unique identifier
- Subject unique identifier
- Extensions field(s)

It is useful to understand the structure of the X.509 certificates. There are several versions of the X.509 standard. The latest version is the X.509v3.

3. ISSUES FOR DESIGNING MAS FOR MOBILE SECURITY

The first step in constructing a methodology must be to define exactly what a methodology is. When constructing a methodology for the creation of Multi-agent Systems, additional effort must go toward consideration of the characteristics of such systems and what that means for the methodology.
Definition of a Multi-agent Methodology

The concepts described in the previous section are consistent with many design methodologies such as the object-oriented methodologies described by Pressman (Pressman 1992). Agents are in much respect extensions of objects, therefore, object-oriented languages are the languages of choice for agents.

Definition of a Methodology

All software-engineering processes require the use of a methodology whose main role is to identify the requisite steps that permit us to proceed from the project requirements to implementation. In other words, a methodology is a guide through the software lifecycle. A methodology provides tools and abstract concepts for transforming a subjective vision of a system (a set of requirements) into an objective formal implementation. It provides software engineers with a map from the original blueprint to final code. Authors like Drogoul and Collinot discuss general traits of methodologies and propose required characteristics of MAS methodologies in their case study of robotic. They state that in general, a methodology contains:

1. A structured set of guidelines, steps, advice for the steps, and how to proceed from one step to the next
2. A unified documentation procedure
3. Consistent use of terminology which has a meaning at each step in the cycle
4. The use of conceptual abstractions
5. A comprehensive history of the project for backtracking purposes

The creation of MAS focused on all of the listed items, with an emphasis on the backtracking mentioned in number five. However, by only considering the listed items, the particulars of Multi-Agent Systems would not be figured into the methodology. However, MASs created using object-oriented methodologies will not take full advantage of agent characteristics. The additional characteristics proposed by Drogoul and Collinot as required for any MAS methodological framework are:

1. Integration of the descriptive and operational aspects of the agent organisation during the analysis phase.
2. The possibility of combining a bottom-up approach (designing agents before organisation) with a top-down (organisation before agents).

Other sources provide similar ideas about the characteristics of a MAS methodology.

The first point above is echoed by Wooldridge, Jennings, and Kinny who say that current software development methods (such as object-oriented) will be unsuitable for MAS design as they fail to capture an agent's flexible autonomous behaviour, rich interactions, and complex organisational structure and also echo the importance of a system-Level or organisational concept.

Drogoul and Collinot's second point, combining top-down with bottom-up, is important in a methodology to ensure a designer has freedom to iterate through different phases of the methodology. For example, the details of a particular system could be specified either before or after a system-Level organisation exists.

MAS considers both of the above listed points. The structure of the organisation is reflected in the ways that roles and agent classes interact through paths of communication and conversations respectively. Furthermore, the phases of MAS are designed to be iterative in support of the second point, and the transformations from goals to roles to agent classes can be followed backward as well as forward.

Means of authentication and authorisation

In this section the actual methods or processes that are used to authenticate the identities of users are discussed. The authorisation of the user to gain access to services or resources can be carried out in a system after the user has been authenticated and his identity is resolved through the use of access control lists (ACL), to determine what the particular user is authorised to do.

Authorisation is thus at maximum as accurate and correct as the process of authentication. A mechanism like the SPKI could be used, to avoid authentication of the user, but to still provide a reliable authorisation.

Passwords

Passwords associated to user names (something that the person knows) are a simple way of authentication. They can have a one-to-one or a on-to-many relationship to actual persons, i.e. every user of a system can have a different password, or a group can have the same password. Implementing a system where passwords are used to authenticate the users is quite straightforward - there just has to be a repository, which should be secure, where the passwords are stored. The disadvantage of using passwords is that they are notoriously vulnerable to attacks. At least the following types of attacks are possible: external disclosures, guessing, communications eavesdropping, replay attacks and host compromise [5].

There are several authentication schemes that make use of passwords in combination with some other factor. A simple extension of passwords is one-time passwords. A user is provided with a list of passwords through a secure channel - e.g. registered post. He only uses each individual password one time to authenticate himself with the system. The use of this scheme already eliminates the possibility of communications eavesdropping and replays attacks.

Password with a token

Passwords can be used in combination with some physical object (something that the person owns). E.g. an ATM card with a PIN code is an example of this - the card itself is useless without the PIN, as is the PIN without the card.

This concept has been extended with the use of integrated circuit cards (ICC) or smart card. They are tamper proof devices that may interact with a device directly, as is the case with e.g. the SIM-card in the GSM-phones. The SIM cannot be used without the knowledge of the PIN-code, which the SIM-card verifies itself. The PIN related to the SIM is never revealed from the SIM. A challenge and response method is used in authenticating the user.

'Synchronous one-time passwords' [5] is another similar technique. The user has a tamper proof device that produces codes that can be used as passwords, often in combination with a PIN and a user name to produce an authentication triplet. The code is produced in the device by an algorithm known also by the authenticator, which thus knows which code has been produced this time and can carry out the authentication. This is a more elaborate scheme than just using lists of one-time passwords, since the use of the device eliminates the need to distribute the lists.

Biometrics

Biometrics authentication techniques include fingerprint recognition, retinal scanning, handgeometry scanning and handwriting and voice recognition [5]. These techniques are all based on the physical properties of a person (something he owns / is).

As the physical properties of different individuals differ sometimes only slightly from each other, but can differ from

time to time on a single individual, it is sometimes impossible to get a high enough level of certainty in the authentication. Furthermore the precision of technologies available is limited and some of the technologies are very costly. In some applications these techniques are perfectly suitable, but in open environments they are still often too difficult to implement into practice.

Digital Signatures

When a PKI is put in place, digital signatures can be used to authenticate users. The following sequence of actions has to be carried out in order to authenticate a user by his digital signature:

1. The user requests access to the service or system
2. The system generates some data for the user to encrypt using his private key. Then the data is sent over to the user.
3. The user concatenates the data received from the system and a time stamp and encrypts the whole sequence. (N.B. It is a good practice that e.g. a time stamp is concatenated to the data, so that the data to be encrypted cannot completely be decided by some untrusted party. This is to avoid the possibility of a 'Chosen plain-text attack' as described in [12].) Then the encrypted data (the cipher text) is sent back to the system. Along with the encrypted data a link to the certificate (or the certificate itself) of the user is sent.
4. The system decrypts the received information with the public key of the user, found in the certificate.
4. The system verifies that the decrypted information is composed of the originally generated data and a valid timestamp. If this seems to be OK, the system has successfully authenticated the user.

Properties of a good authentication and authorisation mechanism

This section lists the properties that a good identity authentication and authorisation mechanism should possess. Some of the features listed are in contradiction of each other, but mostly it should be possible to reach an acceptable Level of compliance with each of the criteria.

Correctness - The results of each individual instance of authentication or authorisation carried out should be correct. If it is possible to authenticate the user, the result should always be that either it is found, that the user is who he claims or he is found to be a fraud. Based on this perfectly correct authentication it is further possible to authorise the user to access those services and resources that defined to be accessible for him or to the group or groups he belongs to. In practice it is impossible to get an absolute certainty in authentication. Only a reasonable Level of certainty can be gained.

Possibility to anonymity and privacy - Identity authentication should only be done when absolutely necessary. Whenever authorisation is possible without the users identity being revealed, it should be done that way.

Speed - The process of authentication should be fast. The user shouldn't have to wait for the result for more than a second does or two.

Attack resistance - The perfect mechanism of authentication should be resistant against any known or unknown types of attacks.

Inexpensiveness - The mechanism shouldn't require extensive investments from either the users or the authenticators.

User friendliness - The mechanism should produce as little overhead to the user as possible. It should also be as easy to use and understand as possible. In the optimum situation the user doesn't have to perform any actions in order to become authenticated. The user shouldn't be forced to carry around any extra equipment, magnetic or smart cards, lists of passwords or other physical objects in order to use the system.

Universality - It should be possible for the user to use the same means or method of authentication in all services and everywhere.

5. A MOBILE USER PROTOCOL

In this section we suggest a general protocol for users authentication, which was propose by the authors in [15] that allow its application into a mobile communication networks. This protocol proposing a new user public-key authentication scheme, based on the digital-signature system modification which employs certificates and elliptical curves, improving with this deficiencies of the secret key authentication protocol; having a low computational complexity and offering a higher security level.

Initial Assumptions

We assume that, when accessing the network in the origin domain, the mobile user is authenticated with a traditional server-based authentication mechanism. Users of every network domain are registered with that domain's Authentication Server1. The AS of a domain can be replicated or partitioned within the domain but the set of all partitioned and duplicated AS´s represent a single domain level authority. An important characteristic of mobile environments is the speed at which users move across domains in the network.

In addition, taking into account, mobile communication system, compared with the fixed network, has some different properties, such as:

Mobile station moves from cell to cell. This requires that the authentication speed should meet the requirement of real time communication.

The illegitimate access to the network is a top concern. Because it can affect the correct counting of the network. In some system, the authentication of the network to users is not considered.

The computing resources are asymmetric. For example, at the user's side, an eight-bit microprocessor is always used. But at network side, a large-scale computer can be adopted.

Facing these properties, we propose a new authentication protocol such as use the following design criteria.

Design Criteria

The solution must take into account the following design criteria:

Domain separation. Domain specific secret or sensitive information such as the user's secret key or password should not be propagated from the home domain to a visitor domain or between visitor domains.

Transparency to users. Authentication in visitor domains should have minimal impact on the user interface with respect to authentication in the home domain.

User identity confidentiality. It is often desirable to keep both the movements and the current whereabouts of

[1] AS (Authentication Server)

mobile users secret. For his reason, all user identification information must be protected from disclosure.
Minimal overhead. The distance between the home and the visitor domain may be very large. Hence, the number of messages exchanged between the home domain and the remote domain for the purpose of authentication should be kept minimal.

Protocol's fundaments
According with the design criteria and properties of communication mobile systems, we created the following fundament of the protocol, which one, will be describe:
The following notation is used in the protocol and throughout the rest.

User's attributes

Cert A	Authentication certificate
α	Base of the D-H2 discreet logarithm problem
m´S	Module of D-H
A´S	Secret session key
mA	Module of AS
pA	Prime Number to generated keys AS
qA	Prime Number to generated keys As
np	Public key of KCC3
mn	Module of the PK4 of KCC
iddR	Identification of the remote domain
idA, idB	Identification of the users A y B
mα,	Large prime for digital signature or an elliptic curve parameter
NR	Nonce generated by visitor domain
ASR	Authentication server of the visitor domain

KEA[5] Procedure
In the Diffie – Hellman scheme, Alice combine the Bob's public key with her own secret key to create a session key. Then Bob combine his own secret key with Alice's public key to create the same session key. KEAS do this in a different way. Alice and Bob have their secret and public long-term keys, but also, they generate one time secret and public keys for specific session. Alice combine her private long term key with the Bob's session public key, and her private session key with the Bob's long term public key.
Now, we will describe the general procedure that includes a pre-calculus, user formation and user Security (level). It is important to distinguish that the digital signal procedures will be implemented in a user-to-user communication environment. Now, we present the functions for key generating, sign generating and sign verify employed in DSA.

Key generating
Select a prime number q: $2159 < p < 2160$
Select a prime number p: q/p-1
Obs: DSS indicate that p is prime: $2159+64t < p < 2160+64t$:
$0 <= t <= 8$
If $t = 8$, then p is prime of 1024 bits.
Select $h \in Z^*p$ and calculate $g = h\ (p\text{-}1)/q \bmod p$, and repeat until g=1.
(g is the generator of the unique cyclic group with order q in Z^*p)
Select a random integer into the interval [1, q-1]
Calculate $y = gx \bmod p$: (p, q, g, y) are public keys, x is the secret key.

Sign generating.
To sign a message, each entity has to do:
Select an integer $k \in Z$: $k \in [1, q\text{-}1]$, k random
Calculate $r = (gk \bmod p) \bmod q$
Calculate $k -1 \bmod q$
Calculate $s = k -1 \{h(m)+Xr\} \bmod q$, h is the SHA-16 function
If $S = 0$ go to 1
The sign of the message (is the pair (r, s))

Sign verify and validation.
For the signs verify *(r, s),* the *Rx*:
1. - Obtain an authentic copy of the "*Tx*" public key *(p, q, g, y)*
2. - Calculate w = s-1 mod q y $h(m)$
3. - Compute $u1 = h(m)\ w \bmod q$, $u2 = rw \bmod q$
4. - Compute v = (g $u1$ $yu2 \bmod p$) mod q
5. - Accept the sign when and only when $v = r$.

For the user-to-base station communication environment, we must to pay attention to the next constraint
1. A and B choose random keys RA y RB respectively and form a D-H αRx mod mα,
α and mα can be public.
The module mα can be smaller than another module. The session module and the public key module of certificates.
This part is executed through threshold definition.
mα can be shape by elements of KCC
2. - Its shaped an info-field A = (idA, ASα, *(iddR), idB, BSα, *(iddR), TE,CertAA)*(iddR) means that could be an encryption function.
Message shape
S ={ {CertAA }, {Ex, DSAS {S, infox, D-H mn }}}
3. - Protocol security

The protocol idea is that D-H scheme guarantee the security (confidentiality) DSA guarantee the authentication.
In masquerade boundary, a enemy can add to the message, generated by origin entity, but due to the information is signed with the session secret Key, as soon as the origin certificate of the user, result impossible masquerade all of this steps.
Execution and basic protocol
The basic protocol is depicted in figure 1. After initialising for network and mobile users it enables a traveling user to establish a temporary residence in a visiting domain by requesting the transfer of location-dependent authentication references from the authentication server of his home domain to its peer in the visitor domain.

6. MULTI-AGENTS SYSTEMS FOR MOBILE AUTHENTICATION

The Multi-Agent System (MAS) methodology, takes an initial system specification, and produces a set of formal design documents in a graphically based style. The primary focus of MAS is to help a designer take an initial specification of an agent system and actually produce agents in code. This methodology forms the basis of Security Autonomous agentTool development system, which also serves as a validation platform and a proof of concept. The agentTool system also incorporates concurrent and future thesis research that specifies individual agents and protocols, verifies agent communications, and produces a Java implementation of a MAS. Currently, agentTool implements three of the seven

[2] D-H (Diffie-Hellman)
[3] KCC (Commutation Center Key)
[4] PK (Public Key)
[5] KEA (Key Exchange Algorithm)
[6] SHA (Secure Hash Algorithm)

MAS phases.

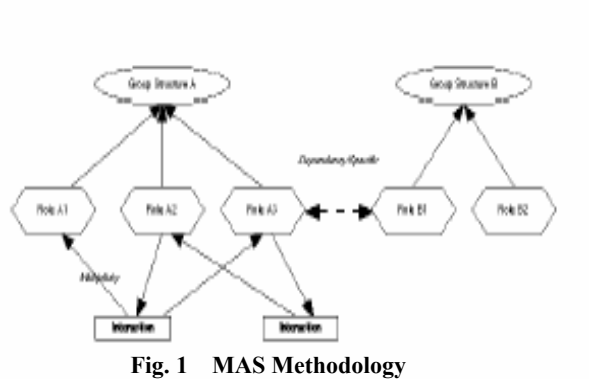


Fig. 1 MAS Methodology

The MAS methodology is independent of a particular system architecture, programming language, or Message-passing system. A MAS designed in MAS could be implemented in several different ways from the same design. The future of agentTool should demonstrate this independence by enabling a MAS design to produce code in several different languages including C++ and Java.

The methodology is similar to traditional software engineering methodologies, and specialised for use in creating Multi-Agent Systems. The general operation of MAS follows the progression of steps shown in Figure 2, with outputs from one section becoming inputs for the next. In practice though, the methodology is iterative across all phases with the intent that successive "passes" will add detail to the models described later. These phases form the next sections of this paper and will be detailed in order. Figure 2 is a simplification of Figure 3 from above that also included all of the data structures involved in MAS.

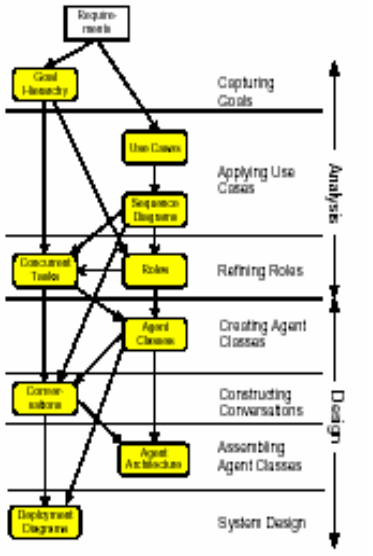


Fig. 2 Organisational Structure for MAS and Mobility

A strength of MAS is the ability to track changes throughout the process. Every design object can be traced forward or backward through the different phases of the methodology and their corresponding constructs. In this manner, backtracking can be performed to find the initial requirements that a particular agent supports. Furthermore the opposite is true; an early-phase object like a goal can be mapped to a set of later-phase objects. The purpose is to eventually be able to select a design object in agentTool and receive visual feedback on all other objects it affects.

Much of the current research related to intelligent agents has focused on the capabilities and structure of individual agents. However, in order to solve complex problems, these agents must work co-operatively with other agents in a heterogeneous environment. This is the domain of *Multi-agent Systems*. In Multi-Agent Systems, we are interested in the co-ordinated behaviour of a system of individual agents to provide a system level behaviour. The methodology can also be successfully applied with traditional software implementation techniques as well. In this research, we view MAS as a further abstraction of the object-oriented paradigm where agents are at an even higher Level of abstraction than typical objects. Instead of simple objects, with methods that can be invoked by other objects agents co-ordinate their actions.

The Methodology Dimension

The description of the methodological framework of a methodology plays a crucial role in MAS design. This description, which is often missing or more or less hidden in the actual MAS design methods, will allow the designers to navigate according to a specific viewpoint when they will be applying the methodology.

To the five criteria retained in [11], two criteria were added: *Availability of Software or Methodological Support* (the existence of a tool supporting a methodology makes it more operational), and *Reusability of Models* (what makes a methodology generic). Thus there are seven (7) criteria for this dimension:

1. *Process Phases*: They are the traditional phases and activities of the software engineering development process. This criterion identifies the various phases and activities that exist in the development process of a methodology. In [11], this criterion is entitled "concerned phases", but we preferred *Process Phases* instead. These phases and activities are: analysis, modelling, specification, design, validation, checking, and ergonomic evaluation.
2. *Development Models*: They are the software engineering models on which the development cycle of a methodology is based. In [11], this criterion is called "development cycle". The models are: cascade model, V model, spiral model, incremental model, and Nabla model [11].
3. *Development Approach*: This criterion identifies the approach followed by the methodology. The development approach can be: top-down, bottom-up or evolving.
4. *User Implication Degree*: Users' contribution usually is decisive in the outcome of a software development project, especially to avoid semantic errors during requirement specifications. Minimising such errors will help keep software development cost to a minimum. This criterion indicates whether the methodology provides means to support and facilitate communication between designers and users. We modified the values of this criterion because we consider that, in any system design methodology, the user must intervene at least to deliver her requirements to the designer. These values are: weak (the user intervenes only at the beginning and at the end of the project), medium (the user also intervenes in the middle but not in all the system development phases), strong (the presence of the user is felt throughout system development).
5. *User Implication Moment*: Beginning (intervenes before the analysis phase to deliver her requirements), middle (intervenes during the analysis, specification and design phases), end (after the design to evaluate the pre-final product).

6. *Models Reuse*: This criterion indicates whether the methodology provides or makes it possible to provide a library of reusable models.
7. *Availability of Software or Methodological Support*: This criterion indicates whether there are available tools that support the methodology (agents libraries, agents components, organisations, interactions, environment, technical or methodological support).

The Representation Dimension

After requirement analysis, a representation of the system from an external viewpoint is normally built. This representation, at least as used by engineers and system architects, will tend to be visual and graphical in order to facilitate comprehension, interpretation and communication between experts, on the one hand, and experts and users on the other hand. Representation is a vital phase in system development, particularly, in Multi-Agent Systems. Control of system complexity requires suitable assembly means (decomposition/construction) in analysis and representation phases. The criteria *System Division* and *Formalisms* of [11] were retained within our framework. We added two other criteria, namely *Models Quality* and *Derivation*. This dimension describes the principles and formalisms used during the methodology's modelling phase. The four criteria are:

1. *System Division*: This criterion indicates the assembly means used during analysis. These are typically carried out by: abstraction levels, generalisation-specialisation, type occurrence, strategy-tactic (to study long-term strategies and the system's tactics).
2. *Formalisms*: This criterion describes the formalisms (diagrams, concepts and rules) used by the methodology. We are especially interested in the formalisms used for the data, the activities, the processes, and the system's dynamics.
3. *Derivation*: A methodology should provide sufficient context to support model derivation and the relationships between these models. This allows system designers to complete their design tasks faster. This criterion indicates the derivation order and the relationships between the methodology's models.
4. *Models Quality*: This criterion can be expressed with several elements such as: the number of models in the methodology; models cohesion; models completion (the models should account for the totality of the problem discourse universe); models complexity (interrelationships between the models, numbers of elements); etc.

The Agent Dimension

This dimension is particularly important for agent-based Systems. It makes possible the description of agents' specific nature, and the characteristics available in the MAS to which the methodology can be applied. The system's performance, efficacy and efficiency are related to these characteristics. Agents' features constitute a determining factor in their social and co-operative behaviour. The ultimate goal would be to endow the agents of anticipation and planning capacity allowing them to optimise their behaviour. Four criteria were associated with this dimension. We added the criterion value Deliberative Behaviour to the *Agent Attributes* criterion. The criteria are the following:

1. *Agent Nature*: Homogeneous or heterogeneous.
2. *Agent Type*: The principal types we retained are: intelligent agents, interfaces or personal assistant agents, mobile agents (e.g. a computer virus), information agents (e.g. a monitor of Web pages, and autonomous agents (responsible of their behaviour).
3. *Agent Attributes*: This criterion indicates the agents' intrinsic characteristics that the methodology uses: adaptability (ability to learn and improve with experience); autonomy (goal directness, proactive and self-starting behaviour); co-operative behaviour (ability to work together with other agents to achieve a common goal); inferential capability (ability to act on abstract task specifications); "Knowledge-Level" communication ability (ability to communicate with other agents with language resembling human-like speech acts instead of simpler symbol Level program-to-program protocols); mobility (ability to migrate in a self-directed way from one host platform to another); personality (ability to manifest attributes of a "believable" human character); reactivity (ability to selectively sense and act); temporal continuity (persistence of identity and state over long periods of time); deliberative behaviour (ability to decide in a deliberation).
4. *Agent Attributions*: How to predict an agent's behaviour without knowing its internal structure? What representations, associated to an agent, best define its observable (external) behaviour and its expected behaviour? The three stances of [12] try to cover these representations types based on: physical laws (being given environment and agent characteristics, one can expect it behaves in several ways); design entity (the inputs/outputs can be understood but not the specific productions); and the intentional stance (the predictions are based on the environment assumptions).

In order to manage the dynamics of the organisations in the SMA, it is necessary to endow each agent with a module of recognition or intention attribution. Recognise the intention of others allows co-ordination, guides the co-operation and, finally, ensures the relevance (with direction of efficacy) of the interactions. Does studied methodology have an attribution's model or provides it mechanisms of attribution?

The Organization Dimension

According to [13], we can define an organisation as a structure describing how the members of the organisation are in relation and interact to achieve a common goal. The structure of an organisation is related to the environment in which it evolves the resources available to produce the outputs as well as the nature of these outputs. MAS construction requires an underlying organisational structure in order to be able to control system complexity. This dimension defines the structure the system must have and the environment's characteristics for which the system is intended. This dimension indicates whether the methodology specifies this structure and these characteristics. The three criteria of [11] were retained in our proposal, but their values were adjusted in order to take into account the architecture and the properties of MAS. We have add Type of Environment:

1. *Organisation Image*: Indicates whether the methodology can represent organisation types (defined by a variety of roles and relationships between these roles). We retained four types amongst those proposed in [11], which are, in our opinion, the main types useful in MAS: hierarchical system, distributed system, open system, holonic system (an architecture resulting from the combination of hierarchical and heterarchical structures).
2. *Environment Nature*: This criterion indicates the characteristics of the environment on which the agents act. These characteristics will allow a cognitive agent to have an explicit representation of its environment and of other agents.

Thus, it will be able to regularise and optimise its behaviour according to the predictions made on the behaviour of the other agents. It is for this reason that we have added the value *explicit* to the three other values retained of [11]. However, an agent cannot posses complete knowledge of its environment.

Does the methodology take into account such considerations during the design of the agent architecture? The values of this criterion are: structured (indicate whether the roles, the functions are clearly defined), stable, determinist (an environment for which starting from its current state, and of the actions which can be carried out there, its future states can be predicted by an informed observer), explicit and observable (an environment for which it is possible to determine all the possible states at every moment, otherwise it is partially observable).
3. Type of Environment: Which can be active and/or passive.
4. *Nature of the Data*: A methodology that can handle equally well numerical and symbolic data is polyvalent and, thus, advantageous.

The Co-operation Dimension
Co-operation is one of the more important aspects of MAS when the agents must cooperate to achieve a common goal. Several authors have studied the influence of the agents' social behaviour on system performance. They have shown that co-operation between agents accelerates the problem resolution process and improves the quality of the solution or the results. It seems highly desirable to find in a methodology general co-operation principles applicable in a consistent fashion and independently of any system particularities. For example, these principles should help establishing and maintaining co-operation between the system's agents. These principles should also make it possible to resolve non-co-operative states such as: incomprehension, ambiguity, competition, conflict, etc. Six criteria were associated with this dimension. The Communication criterion of [11] was retained here and was named *Communication Mode*.
The criteria of co-operation dimension are:
1. *Possible Types of Communication*: Communication between heterogeneous agents or between agents and humans.
2. *Communication Mode*: Which can be direct (sends message), indirect (mail by the post office for example), synchronous (by telephone for example), and asynchronous (electronic mail for example).
3. *Communication Language*: This criterion indicates whether the language used by the agents is based on signals (i.e. low-level language), speech acts or other means.
4. *Co-operation Model*: This criterion indicates the co-operation concepts used in the methodology's interaction models: negotiation (to manage an acceptable agreement for all the agents concerned) tasks delegation (facilitators), or multiagent planning (development and execution of the possible plans).
5. *Control Type*: This criterion indicates the co-ordination type used in the methodology's interaction models: centralised, hierarchical or distributed. In all cases, agents must be able to exchange between them intermediate results and to make transfers of resources. Co-ordination is a central question for the SMA.
6. *Interaction*: This criterion indicates whether interaction is static (e.g. message) or dynamic (e.g. active message). It also indicates whether the interaction engine is distributed (internal with agent) or centralised (server or mediator). In addition, this criterion indicates the nature of the interaction protocols used in the method, which can be explicit or implicit. Let us note that this criterion helped us establishing whether the interaction mechanism used in the methodology was able to resolve non-co-operative states between agents.

The Technology Dimension
The purpose of this dimension is to describe the characteristics of the software on which the methodology is applied. These characteristics constitute, in our opinion, a critical parameter in the choice of a suitable method for a given application. Named *Application Type* here, the Aimed Application Type criterion was added to the three criteria taken from [11] for this dimension. We have added Environment of development criterion. The five criteria are:
1. Processing Mode: They are the possible processing modes of the software at hand: batch, interactive, client-server, synchronous, asynchronous, distributed (distributed modules).
2. Human-machine Interface Type: Classic (when the method does not truly take into account the user interface), adaptable or flexible (configurable by the user as in the VB or VC++ programming languages), adaptive (the interface aims to be adaptable to the user's requirements), and assistant (the interface will reason simultaneously with the user and will behave like a human assistant).
3. Programming: This criterion indicates if it must be structured, object-oriented or agent-oriented.
4. Application Type: Simulation, problem resolution, integration, etc.
5. Environment of development: This criterion describes the characteristics of development of the SMA to which the method can be applied (possible platforms, programming languages, other tools being used to implement the agents, etc).

8. FINAL REMARKS

Because of the great interest in using Multi-Agent Systems (MAS) in a wider variety of applications in recent years, agent oriented methodologies and related modelling techniques have become a priority for the development of large-scale agent-based Systems. Several methodologies have been proposed for the development of MAS. The success of the agent paradigm requires systematic methodologies for the specification, analysis and design of MAS applications. In this dimension we proposed a model in order to achieve mobile security in the issues of Authentication, Authorisation, Public Key Infrastructure and Public-key Cryptography, Certificate Authorities and Digital Certificates

9. REFERENCES

[1] Abelson, H. et al. The Risks of Key Recovery, Key Escrow, and Trusted Third Party Encryption 20.8.1998, [referred 9.11.2000]
[2] Branchaud, M. A Survey of Public Key Infrastructures March 1997, <http://home.xcert.com/ marcnarc/PKI/thesis/>
[3] Diffie, W. and Hellman, M. New Directions in Cryptography, *IEEE Transactions on Information Theory* November 1976, pp. 644-654
[4] Ellison, C. et al. RFC 2693 - SPKI Certificate Theory September 1999
[5] Ford, M. Identity Authentication and 'E-Commerce'
[6] Gerk, E. Overview of Certification Systems: X.509, CA, PGP, and SKIP
[7] Gerk, E. Towards Real World Models of Trust: Reliance on Received Information
[8] [Wooddridge and Jennings 1998] Glaser Norbert 1996 Contribution to Knowledge Modelling in a Multi-Agent Framework (the CoMoMAS Approach), PhD thesis, Université Henri Poincaré, Nancy I, France.
[9] Gustavo A. Santana T., David Higuera R., "A Mobile User Authentication Protocol for Personal Communication Networks, IASTED, WOC 2001. June 2001, Banff Canada.

Java-based PKIX Digital Certificate Authorization Mechanism Design For Internet Distributed Applications

Jiande Lu
School of Computer Science and Technology, Suzhou University
Suzhou, Jiangsu 215006, P.R.China
E-mail: lujiande@public1.sz.js.cn

ABSTRACT

One of the critical issues for Internet distributed applications is the authorization problem. The paper has analyzed the PKIX digital certificate authority mechanism, and has discussed the design and implementation of esPKIX, a platform-independent and Java based Essential PKIX. It covers the design thought in the phases of certificate request, issuance and revocation. esPKIX operates under on-line CA mode and implements the request, issuance and revocation of X.509 v3 certificate on Internet for distributed applications. It also supports the function of CRL v2 and other related functions of administration. Some standard extensions of certificate/CRL and PKCS#8 encoded private key are supported as well.

Keywords：PKIX，CA，CRL，X.509，PKCS，DER，JCA

1. INTRODUCTION

Internet security is very important to an Internet wide distributed system. One of the critical issues for Internet distributed applications is the authorization problem. With a heightened awareness and need for increased security, the usage and implementation of digital certificates has almost become standard. PKI is becoming the core of the enterprise security infrastructure.

ITU X.509 digital certificates are digital documents that bind a public key to an individual or company. The certificates provide the function of authentication and conveyance of information used to establish a secure sockets layer (SSL) connection. In order to trust this relationship between an entity and the public key, it must be avouched. This is done by trusted Certificate Authority (CA), which signs a certificate. The CA includes their digital signature in the certificate thus allowing the certificate holder to confirm they are who they say they are.

At present, the distributed application systems with security function of x.509 digital certificate are getting into more and more use, such as business transaction processing, CSCW environment, electronic commerce and etc. Due to security reasons, we need do thorough research on them. On the other hand, current PKIX(Public Key Infrastructure based on X.509 standard) system performance in heterogeneous environment，especially in various embedded systems, has less suppuort to meet customized needs and have less flexibility in certificate issuance and management. The purpose of this project is to design a essential PKIX (esPKIX) supporting various heterogeneous distributed application and resolve those problems.

The first phase of this project esPKIX operates under on-line CA mode and Java-based, it implements the request, issuance and revocation of X.509 v3 certificate on the Internet for distributed applications. It also supports the function of CRL v2 and other related functions of administration. Some standard extensions of certificate/CRL and PKCS#8 encoded private key are supported as well. This paper will analyze the PKIX mechanism briefly, and discuss the design and implementation of esPKIX, a platform-independent and Java based Essential PKIX developed by us. It covers the design thought in the phases of certificate request, issuance and revocation.

2. PKIX MODEL AND ESPKIX

The concept of a PKI system is based on the use of public and private key pairs. The keys are mathematically related and are used for the encryption and decryption of data. The key pair holder will maintain their private key in strict confidence while the corresponding public key can be freely distributed. Embedded within the X.509 digital certificate is the public key. Certificate Authorities (CA) provide, manage, revoke and renew digital certificates. The main job of PKI is to establish the trust hierarchy between the CA, Registration Authority (RA), and the certificate repository. The major components of PKI are CA, certificate repository, certificate management protocol, registration authority (RA) and end entities (PKI certificate user and/or end user system). This project will focus on designing and implementing essential Java-based PKIX Digital Certificate Authorization Mechanism Design. The overall view of PKI is shown as the figure 1.

PKI processing is logical and often common sense. These processes and procedures are:

(1) An individual fills out a web-based form and submits it. The request is added to a queue of certificate requests.

(2) The Registration Authority (RA) is responsible for confirming the identity and authorization of the individual to obtain a certificate. RA then requests a certificate on behalf of the user. Smaller scale PKIs uses the Certificate Authority (CA) to perform this function. An out-of-band process must be established.

(3) The CA validates the requests from the registration authority. CA signs the certificate with its private key and issues the individual the certificate. CA then publishes the public key to the appropriate LDAP (Lightweight Directory Access Protocol) directories or alternate data stores.

(4) Later, when user requests some controlled resource to application, application will ask user for digital certificate, user enters local certificate database password and selects certificate to present, application will authenticate certificate with LDAP server or CA, meanwhile checking if that certificate is in the Certificate Revocation List (CRL). If the certificate is not valid, the application will decline the access to controlled resource, otherwise application will grant access to that resource.

The main related PKIX protocols are:

- ✧ Certificate and CRL profile [RFC2459]
- ✧ Certificate FTP/HTTP operational protocol [RFC2585]
- ✧ LDAPv2 [RFC1777] and LDAPv3 [RFC2251]
- ✧ Online Certificate Status Protocol OCSP [RFC2560]
- ✧ Certificate Management Protocol CMP [RFC2510]
- ✧ Certificate Request Message Format CRMF [RFC2511]

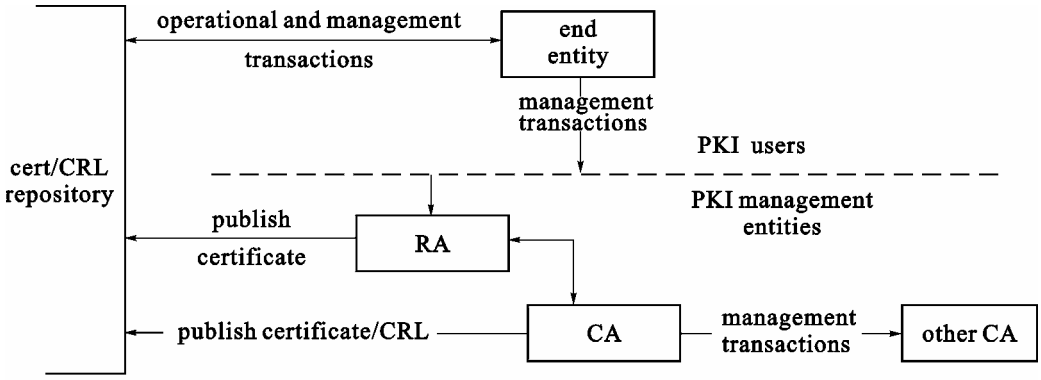


Figure 1 PKIX Architecture

The current commercial PKIX has some shortcomings. For example, the certificate request and management have less flexibility; methods for request are less (some CA only with browser); some CA platforms may have "backdoor"; lacking heterogeneous end entities software; not supporting customizing (extension or reduction); OS-specific and etc. So developing esPKIX is of significance, this is because:
(1) Whole esPKIX will be programmed with Java. It supports heterogeneous end entities. As long as end entity supports Java Virtual Machine, Java-coded end entity software will be "compile once, use everywhere".
(2) More secure, platform is Linux, no "backdoor".
(3) Can be customized to suit various use (extension or reduction).
(4) Java-coded esPKIX can run on multiple platforms, not OS-specific.

3. THE DESIGN AND IMPLEMENTATION OF ESPKIX

The system is programmed with Java and we select encrypt provider JCSI to encrypt the bottom layer data, sign the certificate and provide various security services, so as to get system platform independence and good extension performance.
JCSI implements standards-based security services in a manner compatible with the Java Cryptographic Architecture (JCA), enabling easy integration of these services with the Java Platform. The JCSI suite includes a software JCA/JCE provider for a wide range of cryptographic algorithms and a hardware JCA/JCE provider which wraps around a PKCS#11 library for a hardware token; plus higher-level libraries for PKI, SSL, CMS, S/MIME, Kerberos 5 and XML digital signatures, thereby offering a solution for virtually every data security requirement. The developing environment of esPKIX is Borland JBuilder 6.0 Enterprise Edition.

3.1 Certificate format

In order to make signature, the certificate is encoded with DER. The ASN.1 certificate format is defined as:

```
Certificate ::= SEQUENCE {
    tbsCertificate          TBSCertificate,
                            //not signatured certificate content
    signatureAlgorithm      AlgorithmIdentifier,
                            //signature algorithm
    signatureValue          BIT STRING    }
                            //signature value

TBSCertificate ::= SEQUENCE {
CertificateList  ::=  SEQUENCE  {
        tbsCertList             TBSCertList,
        signatureAlgorithm      AlgorithmIdentifier,
 version     [0]    EXPLICIT Version DEFAULT v1,
                            //version, here is v3
    serialNumber            CertificateSerialNumber,
                            //certificate serial number
    signature               AlgorithmIdentifier,
                            //signature
    issuer                  Name,
                            //certificate issuer
    validity                Validity,
                            //valid period
    subject                 Name,
                            //subject name
    subjectPublicKeyInfo    SubjectPublicKeyInfo,
                            //subject public key info
    issuerUniqueID    [1]   IMPLICIT    UniqueIdentifier
                OPTIONAL,       //issuer unique ID
    subjectUniqueID   [2]   IMPLICIT    UniqueIdentifier
                OPTIONAL,       //subject unique ID
    extensions          [3]   EXPLICIT    Extensions
                OPTIONAL        //optional extensions
            }

Version ::= INTEGER    {    v1(0), v2(1), v3(2)    }

CertificateSerialNumber ::= INTEGER

Validity ::= SEQUENCE {
        notBefore       Time,
        notAfter        Time }

Time ::= CHOICE {
        utcTime         UTCTime,
        generalTime     GeneralizedTime }

UniqueIdentifier ::= BIT STRING

SubjectPublicKeyInfo ::= SEQUENCE    {
        algorithm               AlgorithmIdentifier,
        subjectPublicKey        BIT STRING    }

Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension

Extension ::= SEQUENCE    {
        extnID          OBJECT IDENTIFIER,
        critical        BOOLEAN DEFAULT FALSE,
        extnValue       OCTET STRING    }
```

3.2 CRL format

The CRL is also encoded with DER. The ASN.1 CRL format is defined as:

```
        signatureValue          BIT STRING    }

TBSCertList    ::=    SEQUENCE    {
```

```
        version             Version OPTIONAL, //here is v2
        signature           AlgorithmIdentifier,
        issuer              Name,
        thisUpdate          Time,
        nextUpdate          Time OPTIONAL,
        revokedCertificates SEQUENCE OF SEQUENCE
{
        userCertificate     CertificateSerialNumber,
        revocationDate      Time,
        crlEntryExtensions  Extensions  OPTIONAL
                                    //here is v2
                                      } OPTIONAL,
        crlExtensions       [0]     EXPLICIT     Extensions
OPTIONAL
                                    //here is v2
                                      }
```

3.3 PKCS#10 certificate request

The system supports end entity request certificate with PKCS#10 syntax：

```
CertificationRequest ::= SEQUENCE {
    certificationRequestInfo   CertificationRequestInfo,
                                    //certificate request info
    signatureAlgorithm         AlgorithmIdentifier
      {{SignatureAlgorithms}},    //signature algorithm
    signature       BIT STRING    //signature
}
```

The certificate request info is encoded with DER first, then made signature with subject private key, so before PKCS#10 certificate request, user side need generate his own key pair, instead of generated by CA. The Microsoft Enrollment Control in IE, the Communicator in Netscape, the CertUtil.exe in Windows, or the utility keytool in JDK all support generating key pair even PKCS#10 doc. The certificate request info is defined as below:

```
CertificationRequestInfo ::= SEQUENCE {
    version             INTEGER { v1(0) } (v1,...),
                            //version number
    subject             Name,
                            //certificate request subject
    subjectPKInfo       SubjectPublicKeyInfo
      {{ PKInfoAlgorithms }},   //subject public key info
    attributes          [0] Attributes{{ CRIAttributes }}
                          //attributes defined by PKCS#9
}
```

3.4 Design and implementation of certificate generating and issuing

The certificate generating design is shown as the Figure 2.

The certificate includes the public key of subject. The private key of subject can be generated either on end entity side or on the server side (by CA, then distributed to subject user with out-of-band contact).

3.5 Key pair generation

The esPKIX support key pair generation with RSA and DSA algorithm using encrypt provider JCSI. If extended with other encrypt provider according to JCA architecture, the key pair can also be generated with other algorithms.

3.6 Root certificate generation and publishing

In esPKIX, the root certificate is a self-signed CA certificate and it is a trust anchor of whole PKIX. In the root CA certificate, the issuer name is the same as the subject name. The root certificate is an x.509v3 certificate and conform to the RFC2459 definition. When generating the root certificate, the signature key pair must be generated at first. In default, the system uses 1024-bit RSA algorithm. When CA has its own private key, it then could sign root certificate. EsPKIX publishes root certificate in HTTP/FTP. The format of root certificate is the same as that of the normal x.509v3 certificate except the following extension fields:

- Basic Constraints: cA field set to true,
- Key Usage: nonRepudiation, keyCertSign and cRLSign all set to true,
- Subject Alternative Name is the same as the Issuer Alternative Name,
- Authority Key Identifier extension is not needed.

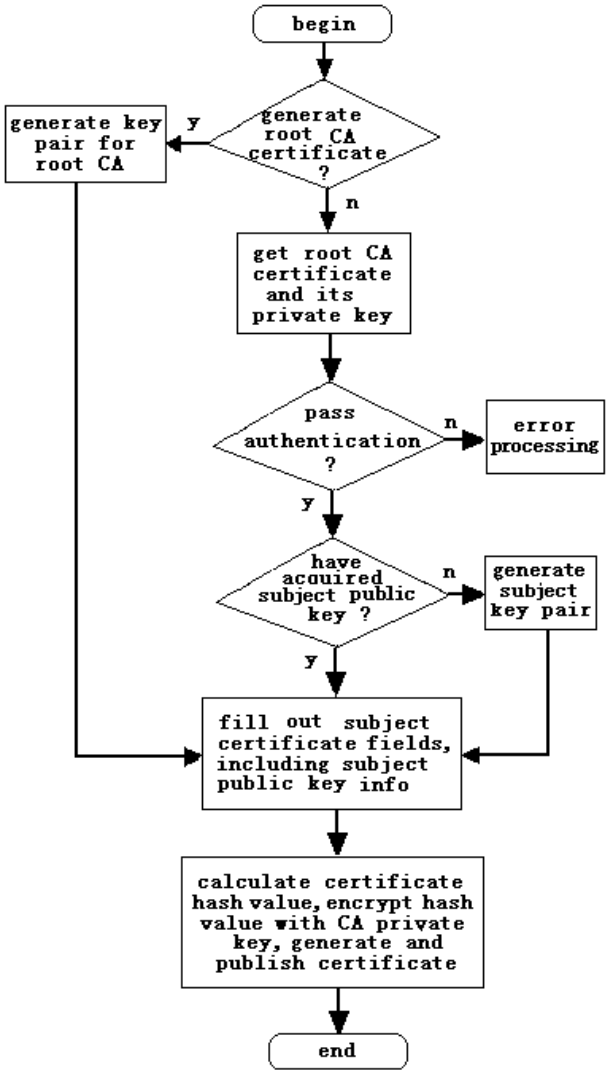


Figure 2 esPKIX CA Processing

3.7 Entity certificate request and publishing

Entity certificates can be classified to subordination CA certificate and end entity certificate. When requesting a certificate, entity/subordination CA needs a private key corresponding to its public key in the certificate, in other words, entity/subordination CA need generate its key pair in advance, say, 1024-bit RSA key pair. EsPKIX is designed to support:

(1) using web browser (IE or Netscape Communicator) to generate key pair on end user side then sending the public key and other info to CA,

or

(2) using Java-coded end entity software utility to generate key pair then sending the public key and other info to CA,

or

(3) generating the key pair for a user on server side then keeping its private key by CA and distributing it to end user in out-of-band method.

(1)(2) are recommended because they would alleviate CA burden and avoid security problems caused by keeping the private key by CA.

When CA generating the entity certificate, CA needs reference his own certificate and private key. System reads some fields info (e.g. organization name) from CA certificate and fills the corresponding fields in entity certificate with this info. When all entity certificate fields are filled, CA signs entity certificate with its private key and generates publishable entity certificate.

EsPKIX uses JSP/Servlet to process certificate request from entity and publish certificate with HTTP/FTP protocol.

3.8 Certificate request with PKCS#10

Certificate requesting with PKCS#10 is quite secure requesting way. After generating key pair on the end user side, esPKIX end entity software encodes certificate request info (subject name, etc.) with ASN.1 DER then signs it with end user's private key, adds subject public key, at last, sends all together to CA.

When CA has received certificate request, at first, CA verifies request contents with subject public key (by comparing the signature value decoded by subject public key with the calculated hash value to certificate request info to see if they are equal, if equal, then pass the validation), then CA generates entity certificate according to request info (subject name, subject public key, etc.). If validation is failed, the request info may be corrupt or tampered by someone in intermediate way, so CA can't generate and publish certificate for that entity user.

3.9 Generating and publishing CRL

When esPKIX CA revokes a certificate, system generates a new CRL (Certificate Revocation List) and need reference to CA certificate and its private key. System fills CRL issuer name field with the content of organization name in CA certificate. It retrieves previous CRL serial number, generating a new serial number sequentially and filling it in the CRL corresponding field. System also fills the serial number of revoked certificate and every extension field. EsPKIX has defined Reason code extension with the default value of certificateHold. When system has filled out all fields, it signs CRL contents with its private key. The generated CRL is published on the Internet with HTTP/FTP protocol.

3.10 Private key saving and PKCS#8

esPKIX's private key saving conforms to PKCS#8 specification. PKCS#8 has defined the syntax format used to encrypt private key. The private key info includes version (0), algorithm, the private key and attributes. EsPKIX system encodes this info with ASN.1 BER and encrypts it with a key using the symmetric algorithm pbeWithMD5AndDES-CBC.

4. CONCLUSION

This paper has illustrated the design and implementation of esPKIX, a platform-independent and Java based Essential PKIX. It covers the design thought in certificate request, issuance and revocation. The first phase developing of esPKIX project has been completed and the running result is satisfying. It can issue certificate/CRL that conforms to series IETF certificate standards and works well in secure E-mail communication and SSL web interaction.

The esPKIX second phase developing will focus on establishing CA chain, LDAP certificate publishing and authentication. When whole project is completed, a more flexible, customized PKIX platform will work under the heterogeneous environment.

5. REFERENCES

[1] Carlisle Adams, Steve Lloyd, "Public Key Infrastructure – Concept, Standard and Implementation", Peoples Post and Telecommunication Press, Beijing, 2001

[2] Steve Burnett, Stephen Paine, "RSA Security's Official Guide to Cryptography", McGraw-Hill, 2001

[3] http://www.ietf.org/rfc, RFC2459, 2585, 1777, 2251, 2560, 2510, 2511

[4] http://www.rsa.com, PKCS#1, #5, #6, #7, #8, #9, #10

[5] http://www.sheca.com

[6] http://java.sun.com

[7] http://www.valicert.com

[8] http://www.dstc.edu.au

[9] http://www.javaunion.org

An Agent-Based Distributed Adaptive Learning Environment

Qiangguo PU*
Computer Center, University of Science and Technology of Suzhou
Suzhou, Jiangsu 215011, China
E-mail: hpu@bigfoot.com

ABSTRACT

The Centre for Computing and Information Systems (CCIS) at Athabasca University, Canada has been developing DALE-an agent-based Distributed Adaptive Learning Environment. This paper presents the agent-based distributed adaptive learning environment-DALE. The author was a senior visiting scholar at CCIS at Athabasca University. He took an active part in the project.

Keywords: agent technology, learning environment, distributed adaptive learning environment, web-based education, e-education

1. AGENT-BASED SYSTEMS

There has been considerable exploration of agent technology applications for education for example: multi-agent approach to the design of peer-help environment (Vassileva et al, 1999) [1]; agents for information retrieval (Hirtz and Wellman, 1997) [2]; agents for student information processing, distribution, and feedback collection (Huhns and Mohamed, 1999) [3]; pedagogical agents (Johnson and Shaw, 1997)[4]; teaching agents (Selker, 1994) [5]; tutoring agents (Solomos and Avouris, 1997) [6]; agents for assignment checking (McCollum, 1998) [7]; agents for student group online support (Whatley et al, 1999) [8]. An empirical study evaluating the effectiveness of intelligent agents in online instruction has suggested that the application of agent technology to online learning hold promise for improving completion rates, learner satisfaction, and motivation (Thaiupathump, 1999) [9]. Lin and Holt(2001) [10] have outlined how distributed intelligent agents can be incorporated to enhance the functionality of DALE (see Figure 1 below). Agents reside on both the student and server machine and support a variety of functions. The architecture is supported by an agent class archive, student profiles (for student modeling) and access to course materials. Student agents reside primarily on the student client after being dispatched from an agent archive on the server.

Functions supported by agents include collaborating with other students, generating tests of random items, tailoring course material to student profile, managing course material presentation, managing external links (reporting new links to of interest to particular students based on their profiles and removing broken links) .

One could distribute the above architecture so that student agents reside primarily on the student computer in a student-learning module (SLM). There may be a role for mobile agents as part of this architecture (Wang and Holt, 2001) [11]. To reduce dependence on a live connection and enhance mobility the bulk of the course materials are distributed to the students' machine along with a student learning module (SLM). A special mobile agent initially resides at the educational institution free the institution. It watches for connections from students. When there has been an update of the course materials since that last connection from a student it then makes a copy of itself, and travels to the student's computer with updating files specific for the identified student. It frees the learner from paying attention to and spending time on updating and maintaining course materials. Mobile agents in the future could also be used for exam invigilation. They even envision learner agents that reside primarily on the learners' computers and visit and carry out a variety of tasks on the hosting servers of a variety of educational institutions.

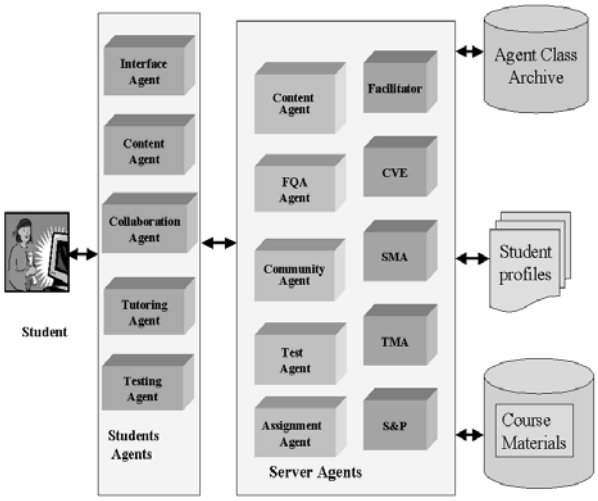


Figure 1 An Agent-Based Architecture for DALE

Figure 2 shows the agent system infrastructure proposed.

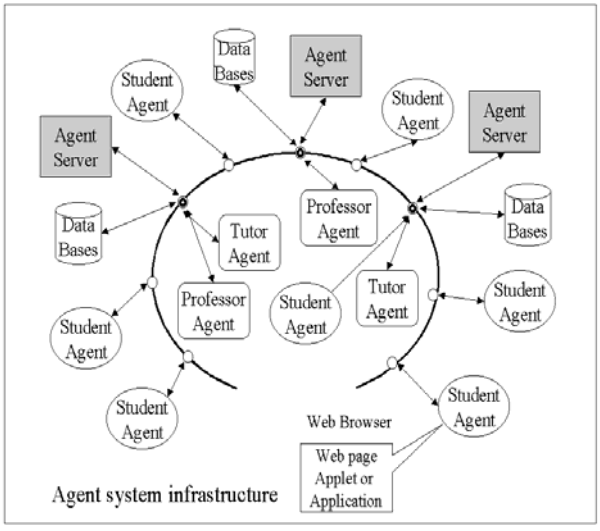


Figure 2 Distributed agent system infrastructure for Web-based learning

2. DESIGN STRATEGIES

A Social-Psychological Human Computer Interface For Learners

Most Human Computer Interaction (HCI) studies are tied to a specific media and device (for example screen aesthetics, the desktop metaphor, and the organization of materials on the screen). There are exceptions to this such as specifying the behavioral and organizational parameters of virtual universities (Holt and Gismondi, 1995) [12]. One can determine how media independent the content of an HCI is by imagining the impact on studies of different topic areas if one changed the media from a graphical user interface to a natural language interface. For most HCI studies drastically changing the media would remove the purpose of the study. However, they think that many of the long-term issues for HCI will be those that are relatively independent of the media of the interface. Specifically, the psychological and social aspects of computers interfaces need to be considered. "Interpersonal relations, group dynamics, personal stress levels, anxiety levels, and productivity…are unexplored" (Hannah, Ball, and Edwards, 1999) [13].

Tracking Standards for Learning Objects and Instructional Design Models

Without standards one will not be able to develop sharable materials, tools, and platforms. The IMS project is developing standards for content packaging so that learning objects can be easily shared. Much research on defining, constructing, and building and displaying learning objects has been conducted over the past few years. The learning objects could be simply online text and images for a particular topic but could also include simulations (Holt, Brehaut, Stauffer, and Jelica, 2000) [14] or VRML (Lin and Ye, 2001) [15]. According to the Sharable Content Object Reference Model (SCORM, 2000), a learning object is modeled as the smallest stand-alone and meaningful component of a course that be interoperable, modular, and discoverable. The CAREO project defines learning objects to include "simulations, tutorials, drill and practice modules, content databases, multi-media exercises" (Downes, 2000) [16]. There remains much work on development of an ontology for subject domains, an ontology for instructional design, definitions of the granularity of learning objects, methods for combining learning objects into courses, and several other issues.

Under the area of instructional design standards the IMS project is trying to develop standards to support various instructional design models such as those discussed previously. It is attempting to develop descriptions for the learning objects and collaborative elements within a design and rules by which learning objects and collaborative elements are recombined in those designs.

3. ANALYSIS AND DESIGN OF THE ARCHITECTURES

We propose a collaborative agent-based system architecture for Web-based distributed learning environment. First, we need to determine agent types and their general architecture. Before determining agent types, we must identify students' needs and tutors' tasks, and professors' tasks in the real world.

3.1 Agent types

From the Web-based educational experience, we know that the students need to

- download course materials
- get dynamic updates to downloaded materials
- connect to obtain and interact with central dynamic materials
- serve materials to other students/tutors
- submit assignment to tutors
- ask tutors for course-related questions
- interact with other students
- have optimal control over their own information/tools etc

Tutors' tasks are

- answering questions from their students;
- knowing student profiles, especially those that need contacting;
- checking students' assignment;
- contacting other course tutors to answer students' questions sometimes.

The primary teaching tasks of professors are to

- develop and update course materials
- serve as course coordinators
- contact to know students' performance and feedback about the courses
- understand student's profiles taking their courses

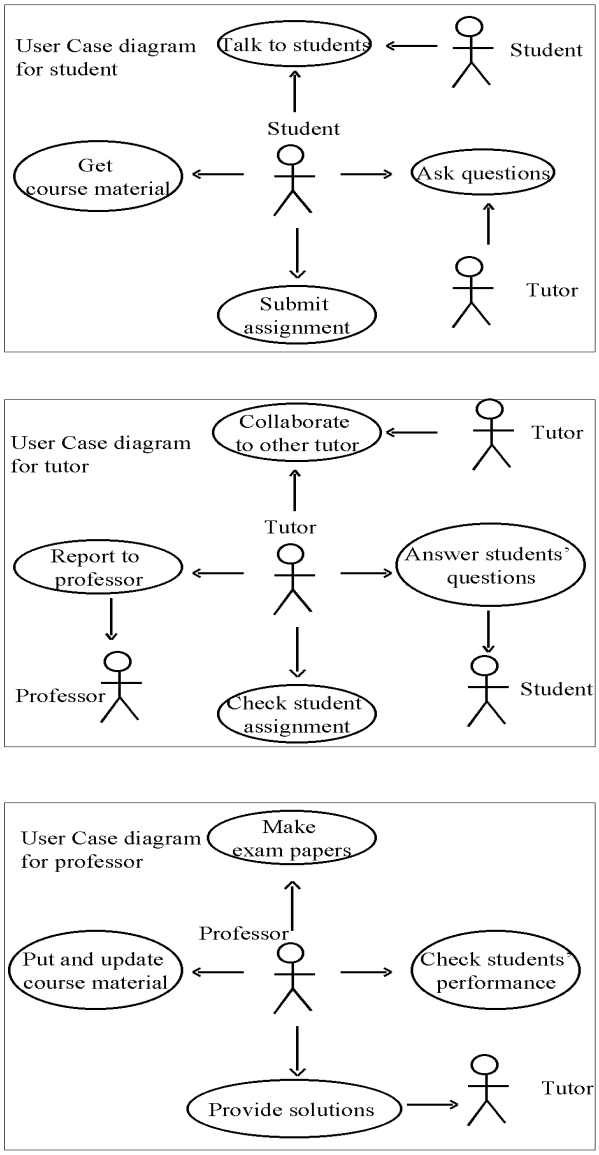


Figure 3 User case diagrams for distributed learning environment

Therefore, we have three types of agents: student agents, tutor agents, and professor agents. These agents' roles are described as follows:

Student agents: Student agents can be hosted by central site for more complex updates. Each student taking an online course will have an interface agent that actually is a collection of more or less independent smaller agents, each having an associated visual presence, downloading initially from the server, operating in the background, watching progress, measuring it against the plan, and taking remedial actions when necessary. These smaller agents are typically some course-related agents. The system includes four course-related agent classes: *content agent* class, *collaboration agent* class, *tutoring agent* class, and *evaluation agent* class. For instance, a content agent has a goal that is to keep the course content updated and fit best to the students. It realizes its goal by continuously monitoring the links for the course and notifying the students taking this course when the links have updated or changed. More importantly, the content agent is able to proactively and adaptively generate the content structure for the student according to the knowledge structure of the course and students' performance. It accomplishes this task by collaborating with a curriculum management agent that is a server agent and is responsible for determining the knowledge structure of a course.

Tutor agents: Tutor agents are responsible for helping tutors to provide on-line tutoring to students. Their role is to interact with student agents to receive and answer students' questions, to do some marking work, and to contact professor agents and other tutor agents within the collaborative learning environments. The tutor agents can further monitor student's interactions [17].

Professor agents: Professor agents' role is helping human professors do course coordination, offer course materials by creating and maintaining course material databases, make examination papers, and offer solutions to exercises of courses. A professor agent may need to inquire into students' learning performance from tutor agents and profiles from students profile database to design adaptive learning materials.

From system point of view, the requirements for the architecture are:

- Distributed: In fact, the Internet is the largest distributed systems. Distributed systems use multiple computers to solve a problem and provide a common, consistent global view of the database system, name space, time, and security, and access to resource. This can be done to increase performance, improve reliability and scalability, or support multiple geographical locations. For instance, to ensure system scalability, in our infrastructure we deploy a set of special agent servers rather than a single one. These servers identify agents that provide services that are the same as or similar to a requested service.
- Mobile agents-based: Multi-agent systems are subject to performance bottleneck in cases where agents cannot perform tasks by themselves due to insufficient resources. A mobile agent approach is much less susceptible to nagging client/server network bandwidth problem, network traffic, transaction volume, number of clients and servers, and many other factors. Furthermore, the downloading of aspects of our course materials gives us a great environment for testing distributed agents and mobile agents. We enhance the existing course materials download system with mobile agent-based course material disseminators. These mobile agents embody the so-called "Internet push model". They can disseminate information such as news and automatic course material updates for course instructors. The agents can carry the new materials as well as installation procedures directly to the students' personal computers. These agents can manage material on the computer creating personalized learning materials.
- Incorporation of Legacy-Systems: For those components in use, our strategy is to incorporate them as far as possible by giving them wrappers. Incorporating these components into different agents' structures and integrating them with new developed components would make our system more cost-effective. Each student agent has currently incorporated a course material manager, a VHD, a WWW Conferencing system, an E-mail system, and a white board. Each tutor agent has been equipped with a VHD, a WWW conferencing system, a white board, and a TRIX, and Course material manager. Each professor agent uses a WWW conferencing, an E-mail, a white board, a TRIX, and a course material manager.

3.2 Agent architecture

Agents in this infrastructure are active, persistent, software components with specialized architecture. All agents in the infrastructure have common architecture even though they have different functionality and knowledge bases. The key requirements that our (tutor, student, professor) agent architecture must satisfy are:

- determine what messages other agents accept and send;
- perceive, reason, act, and communicate with their environment and other agents;
- collaborate with each other, stating from a request for support and continuing by responding to the collaboration-request message in an asynchronous way;
- provide services by using its local knowledge;
- move from one server to other server if needed.

An agent consists of any body of software code whatsoever, with the associated wrapper (see Figure 4). The agent's code consists of a set of *data structure* manipulated by the agent and a set of *functions* that manipulate the data types. A *conversation manager* in an agent is responsible for controlling and coordinating the agent's communication actions to follow conversation protocols. A *mailbox*-based agent communication mechanism is used to avoid hiding the flow of execution and to support autonomy, independence, and the nature of distributed software entities. The *knowledge base* for an agent in the system is implemented as Java Expert System Shell (JESS) [18] files. This has the advantage over embedded procedural knowledge in that there is a clear separation between the inference engine and knowledge required for the agent to function. The *action base* is a set of actions that the agent can use to change its state. The actions can be implemented either logically or through procedure code. The *Concurrent Action Mechanism* component takes as input an agent state and a set of actions, and returns as output as a single action.

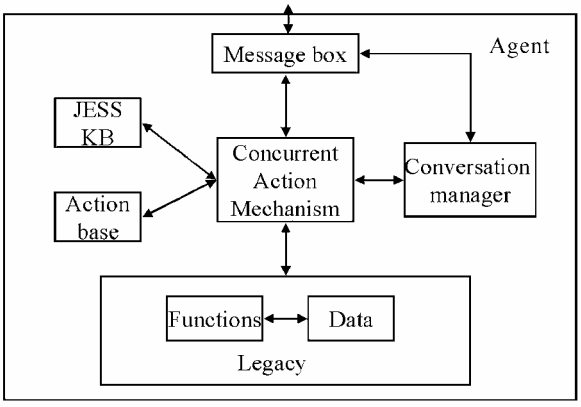


Figure 4 Agent architecture.

Java's security policies (by default) do not allow applets to write into local files. We get around this problem in the following two ways: (1) Use an Applet to implement the GUI for the student interface agents, and make provisions for allowing the Applet to write into local files. (2) Implement the agents as a Java application (stand-alone) program where the GUI is implemented in a Frame.

3.3. Agent Servers

An agent server facilitates inter-agent collaboration and provides yellow page service, registration service, concurrency control, and load-balancing service.

The key requirements that the agent server architecture must satisfy are:

- allow multiple agents to co-exist and execute simultaneously;
- allow agents to communicate with each other;
- be able to freeze an executing agent and transfer it to another agent server;
- be able to thaw an agent transferred from another and allow it to resume execution;
- prevent agents from directly interfering with each other.

In each agent server, we have the following several components:

- a class loader to remotely load agents as Java classes from a secured directory of the agent archive;
- a multithreaded execution environment for agents;
- services for agent operation, including yellow page service;
- services for security, communication, and a proxy for applets, which allows applets running within browsers to operate as agents within agent systems.

The *Concurrency Control and Load Balancing* mechanism of an agent server as showed in Figure 5 is responsible for inter-agent coordination to prevent the agent system from anarchy or deadlock and for time resource load balancing through a load-balancing mechanism. The *Yellow Pages* is a service facilitator that maintains an agent table. The *Thesaurus* lets the owner of a new agent browse a thesaurus and find words similar to the ones he or she is using to describe services. The *White Pages* is for naming and locating agents, including agent management for controlling creation, deletion, suspension, resumption, authentication and migration of the agents. The *Conversation Manager* [19] is for routing messages in Agent Communication Language (ACL) [20] between local and remote agents, based in other agencies. Therefore, the *Agent Platform* provides communication and naming infrastructure using the *Internal Platform Message Transport*.

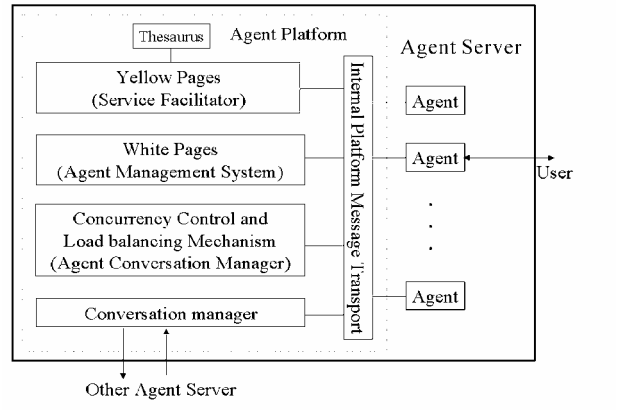


Figure 5 Agent server architecture.

3.4 Agent Deployment

We are developing a Java-based environment that provides secure deployment of agents based upon their roles within the environment. The agent classes are dispatched from a protected agent archive to the agent servers by using a policy-based "single-hop" mobility approach [21]. The agents inhabiting an agent server can be downloaded, as an application, to a local machine of a user or as an applet, running in a client machine of a user via a browser. Figure 6 shows the agent deployment diagram.

In the future we expect that students may run their own servers and perhaps serve and clone student agents to carry out tasks on the central hosts.

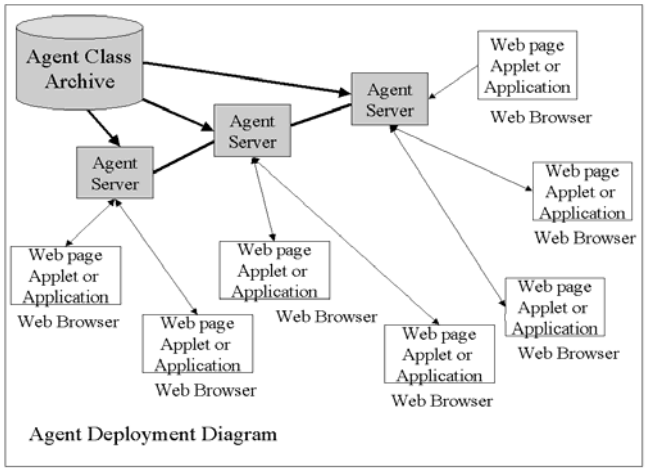


Figure 6 Agent deployment diagram

3.5 Load Balancing

In the workload balancing for software agents, Shehory et. al. (1998) have described a system that uses agent cloning and task splitting as a means for resource allocation [13]. Workloads for agents are balanced by cloning and task distribution. Simulated results showed the cloning technique would be expected to provide an improvement in the effective "throughput" of the agent system once the number of tasks is large enough to warrant the reasoning overhead. In our case, the criteria for successful operation of the agent system are service availability and access speed. In fact, our infrastructure can be viewed as a distributed Web-server system consisting of multiple Web-server hosts, distributed on LANs and WANs, with a mechanism to spread incoming client requests among the servers. Each server in the system can respond to any client request. There are several possible approaches to do load-balancing [14], which must be transparent to users, making a distributed system appear as a single host to the outside world. We use the server-based approach by which distributed scheduling provides scalability without introducing a single point of failure or potential bottleneck. It also achieves the fine-grained control on request assignments as dispatcher-based solutions [14].

4. CONCLUSIONS

We have presented an infrastructure that is a framework for integrating a community of distributed software agents in a distributed learning environment. This framework is designed to function as the foundation of our research development project. We believe it will facilitate flexible, adaptable interactions among distributed components through delegation of tasks, data requests & triggers. It also enables natural,

mobile, multi-modal user interfaces to distributed educational services. The agent system infrastructure proposed has the following features:

- Legacy-incorporated agent architecture
- Multiple agent servers with load-balancing mechanism
- "Single-hop" mobility-based agent deployment approach
- Notification (push) and request (pull)-agent interaction

5. REFERENCES

[1] Vassileva, J., Greer, J., McCalla, G., Deters, R., Zapata, D., Mudgal, C. and Grant, S.. A multi-agent design of a peer-help environment. *Artificial Intelligence and Education, Proceedings IAIED'99*. LeMans, France:IOS Press. 1999.

[2] Hiltz, S. R. and Wellman, B.. Asynchronous learning networks as a virtual classroom. *Communications of the ACM*, Vol. 40, N. 9, pp. 44-49. 1997.

[3] Huhns, M., A. Mohamed. "Benevolent Agents", *IEEE Internet Computing*, Vol.3, No.2, pp.96-98. 1999.

[4] Johnson., W. L. and Shaw, E. Using Agents to Overcome Deficiencies in Web-Based Courseware, *AI-ED'97*. 1997.

[5] Selker, T., Coach: A Teaching Agent that Learns, *Communications of ACM*, 37(7), 1994.

[6] Solomos, K. and Avouris, N. Learning from Multiple Collaborating Intelligent Tutors: An Agent-based Approach, J. of Interactive Learning Research, 10(3/4), pp 243-262, 1997.

[7] McCollum, K. How a Computer Program Learns to Grade Essays, *The Chronicle of Higher Education*, September 4, 1998.

[8] Whatley, J.G, Staniford, Beer, G. M. and Scown, P. Intelligent Agents to Support Students Working in Group Online, *Journal . of Interactive Learning Research*, 10(3/4), pp 361-373. 1999.

[9] Thaiupathump, C., J. Bourne, J. and Campbell, J. Intelligent Agents for Online Learning, JALN Volume 3, Issue 2 - November 1999.

[10] Lin, F., Holt, P. Towards Agent-based Online Learning, *CATE'2001*, Banff, Canada. 2001.

[11] Wang, H., Holt, P. Using Mobile agent to update and maintain course materials on Students' Computers in Web-Based Distance Education, CATE2001, Banff. 2001.

[12] Holt, P., & Gismondi, J. Integrating virtual spaces into open learning systems. *Innovations in Education*. Minot, ND, USA. 1995.

[13] Hannah, K., Ball, M., & Edwards, M.J.A. *Introduction to nursing informatics (2nd. Ed.)*. New York: Springer-Verlag, 1999.

[14] Holt, P., Brehaut, W., Stauffer, K., Jelica, G. & Gismondi, J. A Distributed Web-based Learning Environment for Computer Science Education. *Webnet 2000*, San Antonio, Texas, USA. 2000.

[15] Lin, Fuhua, Ye, Lan. Developing Virtual Environment for Industrial Training, *An International Journal of Information Science*, *Special Issue on Interactive Virtual Environment and Distance Education*. 2001. http://javasim.ncl.ac.uk

[16] Downes, S. Exploring New Directions in Online Learning, 2000
http://www.atl.ualberta.ca/downes/naweb /Exploring.html

[17] Shih, T. K., Chang, S. K., Wang, C. S., Ma, J., Huang, R. An Adaptive Tutoring Machine Based on Web Learning Assessment. In: Proceedings of the IEEE Inter. Conf. on Multimedia and Expo 2000, New York City, USA, July 31 – August 2, 2000.

[18] Java Expert System Shell (JESS). < URL: http://herzberg1.ca.sandia.gov/jess/>

[19] Lin, F. , Flores mendez, R. A., Kremer, R., Norrie, D. H. Incorporating Conversation Managers into Multi-agent Systems, Agents'2000 workshop on Agent Communication, Spain. 2000.

[20] Labrou, Y., Finin, T., Peng, Y. The current landscape of Agent Communication Languages. In: Intelligent Systems, 1999, 14 (2).

[21] Korba, L., Lin, F. Towards Policy-Driven Agent Development and Management, Second International Workshop on Mobile Agents for Telecommunication Applications (MATA) , Paris, France, 2000.

* Qiangguo Pu is an associate professor of Computer Center at University of Science and Technology of Suzhou, China. He was a visiting scholar at McMaster University, Canada and a senior visiting scholar at Athabasca University, Canada. He is author or co-author of more than 40 publications (journal papers and conference proceedings). His research interests include the application of computers and other technology in distance education and control systems.

Adaptive Load Balancing of Parallel Applications with Reinforcement Learning on Heterogeneous Networks

Johan PARENT
COMO, VUB
Brussels, Belgium
E-mail: johan@info.vub.ac.be
Katja Verbeeck
COMO, VUB
Brussels, Belgium
E-mail: kaverbee@vub.ac.be
And
Jan LEMEIRE
PADX, VUB
Brussels, Belgium
E-mail: jlemeire@info.vub.ac.be

ABSTRACT

We report on the improvements that can be achieved by applying machine learning techniques, in particular reinforcement learning, for the dynamic load balancing of parallel applications. The applications being considered here are coarse grain data intensive applications. Such applications put high pressure on the interconnection of the hardware. Synchronization and load balancing in complex, heterogeneous networks need fast, flexible, adaptive load balancing algorithms. Using reinforcement learning it is possible to improve upon the classic job farming approach.

Keywords: Parallel processing, Adaptive load balancing, reinforcement learning, heterogeneous network, intelligent agents, data intensive applications.

1. INTRODUCTION

Load balancing is crucial for parallel applications since it ensures a good use of the capacity of the parallel processing units. Here we look at applications which puts high demands on the parallel interconnect in terms of throughput. Examples of such applications are compression applications which both process important amounts of data and require a lot of computations. Data intensive applications [2] require a lot of communication and are therefore dreaded for most parallel architectures. The problem is exacerbated when working with heterogeneous parallel hardware. This is the case in our experiment using a heterogeneous cluster of PCs to execute parallel application with a master-slave software architecture. Adaptive load balancing is indispensable if system performance is unpredictable and no prior knowledge is available [1].
In the multi-agent community, adaptive load balancing is an interesting testbed for multi-agent learning algorithms, likewise for multi-agent reinforcement algorithms as in [8, 10]. However the interpretations and models of load balancing there are not always in the view of real parrallel applications. We report on the results of adaptive agents in the farming scheme.
The bottleneck for parallelizing data intensive applications is the link to the master. The presented results show that using reinforcement learning it is possible to reduce the strain on the communication hardware. This can be achieved by individually adapting the amount of data (block size) requested by each slave. The learning scheme proves to be better than the relatively efficient sequential job farming scheme.

This document is structured as follows. Section 2 introduces reinforcement learning. Section 3 gives an overview of the existing load balancing strategies. Section 4 presents the experimental setup and section 5 reports the experimental results. The last section concludes with results and future work.

2. REINFORCEMENT LEARNING

Reinforcement learning is the problem faced by an agent that learns behavior through trial-and-error interactions with a dynamic environment. A model of reinforcement learning consists of a discrete set of environment states, a discrete set of agent actions and a set of scalar reinforcement signals. On each step of interaction the agent receives reinforcement and some indication of the current state of the environment, and chooses an action.
The agent's job is to find a policy, i.e. a mapping from states to actions, which maximizes some long-run measure of reinforcement. These rewards can take place arbitrarily distant in the future. To obtain a high overall reward, an agent has to prefer actions that it has learned in the past and found to be good, i.e. exploitation, however discovering such actions is only possible by trying out alternative actions, i.e. exploration. Neither exploitation, nor exploration can be pursued exclusively.
Common reinforcement learning methods, which can be found in [6, 12] are structured around estimating value functions. A value of a state or state-action[1] pair, is the total amount of reward an agent can expect to accumulate over the future, starting from that state. One way to find the optimal policy is to find the optimal value function. If a perfect model of the environment as a Markov decision process is known, the optimal value function can be learned with an algorithm called value iteration. An adaptive version of this algorithm exists for situations were a model of the environment is not known in advance.
For instance the Q-learning algorithm, which is an adaptive value iteration method [6, 12] bootstraps its estimate for the state-action value $Q_{t+1}(s,a)$ at time t+1 upon its estimate for $Q_t(s',a')$ with s' the state where the learner arrives after taking action a in state s:

[1] In control problems approximating action/ value functions is more interesting because then there is no need for knowing the environments transition dynamics.

$$Q_{t+1}(s,a) = (1-\alpha).Q_t(s,a) + \alpha.(r + \gamma.\max_{a'} Q_t(s',a')) \quad (1)$$

With α the usual step size parameter, γ a discount factor and *r* the immediate reinforcement.
In our load-balancing problem setting processors are of the receiver-initiated type and can thus be viewed as agents, which we will give extra learning abilities. Each processor will be an independent Q-learning agent, which tries to learn an optimal chunk size of data to ask the master, so that the blocking time for others is minimized.

3. LOAD BALANCING

Load balancing is assigning to each processor work proportional to its performance, minimizing the execution time of the program. But processor heterogeneity and performance fluctuations make static load balancing insufficient [1]. We investigate dynamic, local, distributed load balancing strategies [13], which are based on heuristics, since finding the optimal solution has shown to be NP-complete in general [9]. Following the agent philosophy, the request assignment strategy is a receiver-initiated algorithm [5], in which the "consumers" of workload look for producers [11]. The goal is a fast adaptive system that optimizes computation and synchronization.

4. EXPERIMENTS

Problem description
In situations were the communication time is not negligible, as is the case for data intensive applications, faster processing units can incur serious penalties due to slower units. A data request issued by a slow unit can stall a faster unit when using farming. This of course results in a reduction of the parallelism. This phenomenon is bound to occur when slave request identical amounts of data from the master. And this independently of the actual amount (we here neglect the communication delay, which is acceptable given sufficiently big requests).
In order to improve upon the job-farming scheme when working with heterogeneous hardware, the slaves have to request different amount of data from the master (server). Indeed their respective consumption of communication bandwidth should be proportional to their processing power. Slower processing units should avoid obstructing faster ones by requesting less data from the master.

Computation model
The initial computation model is sequential job farming. In this master-slave architecture the slaves (one per processing unit) request a chunk of a certain size from the master. As soon as the data has been processed the result is transferred to the master and the slave sends a new request (figure 1).
This scheme has the advantage of being both simple and efficient. Indeed, in the case of heterogeneous hardware the load (amount of processed data) will depend on the processing speed of the different processing units. Faster processing units will more frequently request data and thus be able to process more data.

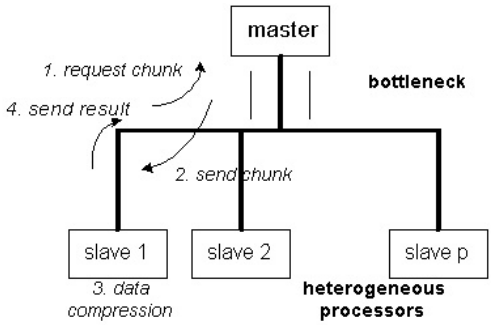


Fig 1 Model

The bottleneck of data intensive applications with a master-slave architecture is the link connecting the slaves to the master. In the presented experiments all the slaves share a single link to their master (through an ethernet switch). In this scenario the applications performance will be influenced by the efficient use of the shared link to the master. Indeed, the fact that the application has a coarse granularity only insures that the computation communication ratio is positive. But it does not preclude a low parallel efficiency even when using job farming.

Experimental setup
To assess the presented algorithm for coarse grain data intensive applications on heterogeneous parallel hardware a synthetics approach has been used. An application has been written using the PVM [3] message-passing library to experiment with the different dimensions of the problem. The application has been designed not to perform any real computation, but instead it replaces the computation and communication phases by delays with equivalent duration[2].

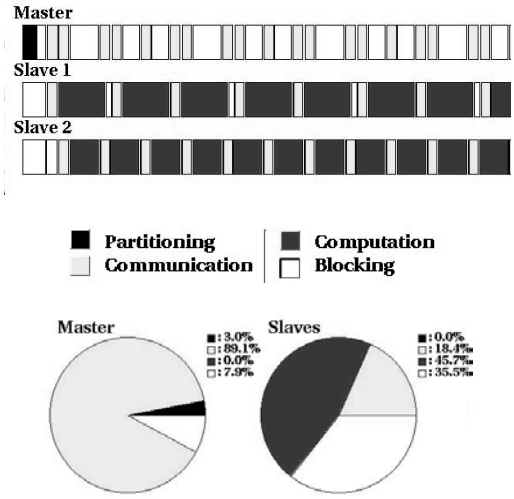


Fig 2 Computation bottleneck

The advantage is the possibility of configuring all the parameters during an experiment, both the granularity of the hardware and the software can be chosen. Choosing the ratio between bandwidth of the network (MB/sec) and the execution speed of the processors (MB/sec) can set the hardware granularity. The software granularity is ratio of the required communication (MB) and processing (MB) and can thus be chosen as well. This gives more control when running the

[2] The experimental application can easily be turned into a real application by replacing the delay producing code with real code.

experiments without losing the essential behavior of such an application.

We will investigate the non-trivial case where the total task computation time is comparable with the total communication time through the bottleneck. This is achieved by setting the average granularity equal to the number of processors.
When the total computation power is lower than the master's communication bandwidth, there is no bottleneck, the master will be able to serve the slaves constantly and these will work at 100% efficiency. This can be seen in the experiment of figure 2, where the total computation power is 0.45 of the total communication bandwidth. But for data intensive application, this won't be the case; moreover, more processors can be added. On the other hand, when the total computation power is higher, the communication at the master will serve as the bottleneck, reaching 100% efficiency. But the efficiency of the slaves will drop, so the surplus of slave processors should better be used for other computations (as in the Grid philosophy). This can be seen in figure 3, where the total computation power is 2.45 of the total communication bandwidth.

Learning to request data
The algorithm is based upon the concept of a Stochastic Learning Automaton, such an automaton serves the purpose of finding optimal actions out of a set of allowable actions [7]. Unlike a reinforcemtent learner model, a stochastic learning automaton only considers one state and uses a $\gamma = 0$.

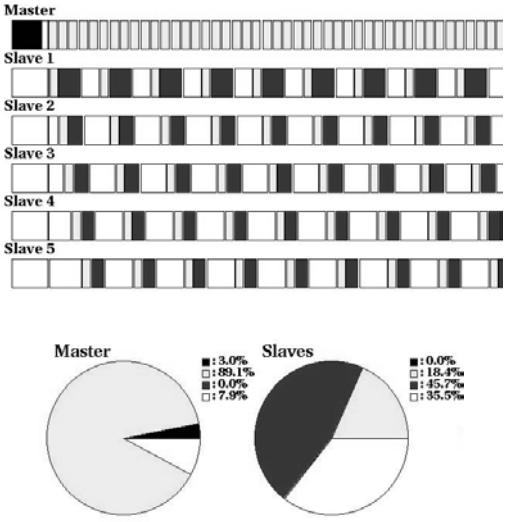


Fig 3 Communication bottleneck

In our experiment, the slaves will learn to request a chunk size that minimizes its blocking time. To that end each slave has a stochastic learning automaton which uses Eq.(1), as shown in Fig 4. In the presented results the block size is a multiple of a given initial block size. Here the multiples are 1, 2 and 3 times the initial block size
The feedback *r* provided to the learner is the waiting time, which is the time a slave has to wait before the master acknowledges its request for data. The inverse of the waiting time (1/ waiting time) is used to update the Q-values using Eq(1). Less interesting actions (i.e. multiples of initial block size) will incure higher waiting times and thus will have lower Q-values associated with them.
The stochastic learning automaton choses an action probabilistically using the Q-values. Which means the actions with a low Q-value have a lower chance of being chosen. In order to get reliable Q-values the stochastic learning automaton present in each of the slaves does not choose a new actions for each request. Instead, each action is performed T times (T=5 during the experiments).

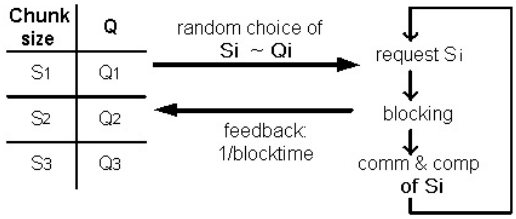


Fig 4 The Reinforcement Learner

We will compare our adaptive algorithm with a static load-balancing scheme where fixed amounts of data are requested.

5. EXPERIMENTS

Figure 5 shows the time course of a typical experiment, with the computation, communication and blocking phases (data size = 800MB, communication speed = 40MB/s, average chunk size = 1MB, processors=4, average granularity = #processors).

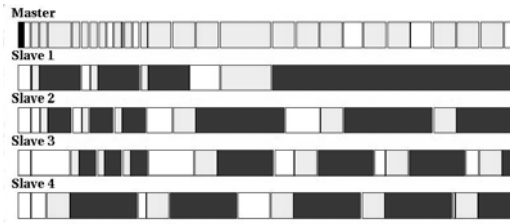


Fig 5 experiment

Performance results
The global goal is minimizing the total computation time, as shown in table 1 (same parameter values as in the previous experiments, we vary the number of processors and keep the average granularity = #processors). In these first experiments, we get an improvement of about 40% of the learner upon the static load balancer.

Table 1 Total Computation Time

#Slaves	Static LB	Adaptive LB	Perf. Gain
4	32.1s	22.6s	-42%
8	32.0s	23.0s	-39%
10	32.2s	23.1s	-39%

Overhead Analysis
Three overheads are responsible for the total computation time (table 2, same experiment as for Fig 5):

- The masters blocking time, which stands for inefficient use of the communication bottleneck at the master.
- The total blocking time of the slaves, due to inefficient synchronization (since the total computation time equals the communication time).
- The total computation time of the slaves, which decreases with better load distribution (better use of the faster processors of the heterogeneous network).

Table 2 Overhead Analysis

	Static LB	Adaptive LB	Perf Gain
Total Computation time	32.1s	22.6s	-42%
Master blocking time	8.2s	5.3s	-35%
Total slave blocking time	80.1s	49.5s	-62%
Total slave computation time	63.1s	54.0s	-16%

The results show that the 3 overheads have decreased, due to better load balancing.
In the next section we will investigate which parameters influence the performance.

Performance Analysis
As can be seen in table 1, the number of slaves doesn't influence the performance gain. On the other hand, the learning algorithm needs a training period. We investigated this by varying the number of requests of the experiments. The results can be seen in figure 6, where we plot the performance gain in function of the number of requests. This parameter was set by varying the blocksize and the number of processors.

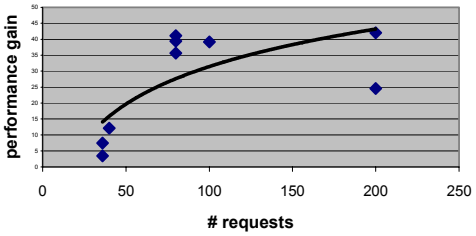


Fig 6 Training Period

6. CONCLUSIONS

Complex, heterogeneous system controlled for optimal use by learning automata is promising. We implemented a reinforcement learner for distributed load balancing of data intensive applications. The first performance results show considerable improvements upon a static load balancer. The algorithm works locally on the slaves (receiver-initiated), thus acting like intelligent agents.
Until now the learning automaton is only fed by local information, which we believe, is too less for fast, highly efficient parallel processing. We will have to extend the feedback by exchanging information between the agents, as explained in the next section.

7. FURTHER WORK

Feeding the reinforcement learner with only local information leads to egoistic behavior or non-cooperative load balancing [4]. Improved performance will be reached by exchanging information between the slaves (or agents), i.e. feedback needed for better synchronization and load distribution.
Another important aspect in heterogeneous networks is the computational contribution of the fast processors, where computation time is of higher value than equal time on slower processors. To take this into account, we first express the processor power relative to a base one [12], which we expressed in our setup by the hardware granularity. Then the processor waiting time, used as feedback for the reinforcement learner, has to be multiplied by this factor. This scales local time with respect to the processor power and will result in a better tuning.

8. REFERENCES

[1] I. Banicescu and V. Velusamy, "Load Balancing Highly Irregular Computations with the Adaptive Factoring", Proceedings of the 16th International Parallel & Distributed Processing Symposium, IEEE, Los Alamitos, California, 2002.
[2] M. D. Beynon, T. Kurc et al. "Efficient Manipulation of Large Datasets on Heterogeneous Storage Systems", Proceedings of the 16th International Parallel & Distributed Processing Symposium, IEEE, Los Alamitos, California, 2002.
[3] A. Geist, A. Beguelin et al., "PVM: Parallel Virtual Machine", the MIT press, 1994.
[4] www.netlib.org/pvm3/book/pvm-book.html
[5] www.epm.ornl.gov/pvm/ (pvm homepage)
[6] D. Grosu and A.T. Chronopoulos, "A Game-Theoretic Model and Algorithm for Load Balancing in Distributed Systems", Proceedings of the 16th International Parallel & Distributed Processing Symposium, IEEE, Los Alamitos, California, 2002.
[7] D. Gupta and P. Bepari, "Load sharing in distributed systems", In Proceedings of the National Workshop on Distributed Computing, January 1999.
[8] Kaelbling L.P., Litmann M.L., Moore A.W.,: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4 (1996) p 237-285.
[9] T. Kunz, "The Influence of Different Workload Descriptions on a Heuristic Load Balancing Scheme", IEEE Transactions on Software Engineering, Vol. 17, No. 7, July 1991, pp. 725-730.
[10] Nowé, A., Verbeeck, K., "Distributed Reinforcement learning, Loadbased Routing a case study", Proceedings of the Neural, Symbolic and Reinforcement Methods for sequence Learning Workshop at ijcai99, 1999.
[11] C.C. Price and S. Krishnaprasad, "Software allocation models for distributed systems", in Proceedings of the 5th International Conference on Distributed Computing, pages 40-47, 1984.
[12] Schaerf A., Shoham Y., Tennenholtz M., "Adaptive Load Balancing: A Study in Multi-Agent Learning", Journal of Artificial Intelligence Research (1995) 475-500.
[13] T. Schnekenburger and G. Rackl, "Implementing Dynamic Load Distribution Strategies with Orbix", International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'97), Las Vegas, Nevada, 1997.
[14] Sutton, R.S., Barto, A.G., "Reinforcement Learning: An introduction", Cambridge, MA: MIT Press (1998).
[15] M.J. Zaki, Wei Li; S. Parthasarathy, "Customized dynamic load balancing for a network of workstations", Proceedings of the High Performance Distributed Computing (HPDC'96), IEEE, 1996.

Congestion Control Method for Real-Time Communications on ATM Networks

Lichen Zhang
Faculty of Computer Science and Technology
Guangdong University of Technology , 510090 Guangzhou
GuangDong Province, P.R. of China
E-mail: lchzhang@gdut.edu.cn

ABSTRACT

ATM Networks are getting increasing popular in supporting real-time application. ATM is expected to be the multiplexing and switching technique for broadband integrated service digital networks which can transport almost all types of traffic including bursty data traffic. Consider real-time applications running on an ATM network, we require the ATM network to provide a transmission guarantee for real-time service. However, this capability cannot be realized without a proper congestion control scheme. When congestions occur in ATM network, the delays encountered by cells begin to increase rapidly, buffers overflow and drop cells beyond a specified limit, and the throughput begins to drop, even approaching zero or deadlock. In this paper, we first review some congestion control methods, then we propose a congestion scheme which is suitable for real-time communication in ATM networks.

Keywords: ATM, Real-Time communication, Congestion.

1. INTRODUCTION

Computer networks are in a period of transition, moving from relatively slow communication links and data-oriented services to high-speed fiber optic links and a diverse set of services. Many of these services, such as voice, video and other applications, will have stringent real-time constraints and but a predictable “quality of service” (QOS) not offered by current best-effort-delivery networks. The Characteristics of real-time communication applications differ significantly from those that are non-real-time. As in real-time computing, the distinguishing feature of real-time communication is the fact that the value of the communication depends upon the times at which messages are successfully delivered to the recipient. Typically, the desired delivery time for each message across the network is bounded by a specific maximum delay or latency, resulting in a deadline being associated with each message.

ATM networks are getting increasingly popular in supporting real-time applications. This network should support a number of communications services, including phone calls, video conferencing and computer communication. However, providing integrated services in high speed store-and-forward networks like ATM is difficult because of the wider range of traffic patterns and quality of service (QoS) requirements to support. Real-time communication services such as video & audio conferencing, video-on-demand, and remote medical services in an integrated network pose serious challenges in meeting their stringent QoS requirements such as bounded cell-delivery delay and cell-loss ratio, while handling the burstiness of their traffic. Real-time communication can be classified into two categories according to QoS requirements: deterministic and statistical. In the former, QoS requirements are specified in deterministic terms and no cell losses or deadline misses are allowed. In order to satisfy its absolute QoS requirements, each deterministic real-time connection requires resource reservation based on the worst-case source traffic-generation behavior, thus resulting in severe underutilization of network resources when source traffic is bursty. In order to make more efficient use of network resources, statistical real-time communication specifies QoS requirements in statistical (instead of deterministic) terms, thus tolerating a certain percentage of cell losses and deadline miss. Such a specification allows for overbooking network resources and, at the same time, enhancing the multiplexing gain. Statistical real-time communication is useful to those applications (1) that can tolerate a portion of cell losses and deadline misses and (2) whose traffic is bursty. The statistical multiplexing gain is substantial, especially in Variable-Bit-Rate (VBR) applications such as MPEG-coded video.

Two types of congestion-control schemes have been proposed: reactive control and preventive control. With reactive control, when a traffic congestion occurs the source nodes are instructed to throttle their traffic flow by giving feedback to them. A major problem with reactive control in high-speed networks is slow feedback. By the time that feedback reaches the source nodes and the corresponding is triggered, it may already be too late to react effectively to the congestion.

Preventive control, on the other hand, does not wait until a congestion actually occurs, but rather strives to prevent the network from reaching an unacceptable level of congestion. A common approach is to control traffic flow at entry points of a network, which is realized with admission control and bandwidth enforcement. Admission control determines whether or not to accept a new connection at the time of call setup. This decision is based on the traffic characteristics of the new connections to ensure that the actual traffic flow conforms to that specified at the time of connection establishment.

In this paper, we first review some congestion control methods, then we adopt priority control algorithm for providing different QoS bearer services can be implemented by using threshold methods at the ATM switching nodes. In the separate route approach, buffer management is not required at the switching nodes, since priority processing is executed at the connection level by the routing function. Priority disciplines in queuing theory can be categorized into two major types: 1). Service or delay priority disciplines, which govern the time at which cells in the buffer are transmitted. 2) Buffer access or space priority disciplines, which govern the input access, or space priority disciplines.

The rest of this paper is organized as follows: Section 2 outlines the characteristics of common ATM real-time services. Section 3 reviews the some existing congestion control methods. Section 4 describes the model of our method. Lastly, we conclude the paper in section 5.

2. ATM REAL-TIME SERVICE

ATM [4] [6] is a connected –oriented packet-switched

technology. Before two hosts began to communicate, a connection is then often to denote the stream of messages sent from a source host to a destination host. In ATM networks, messages from individual connections are segmented into fixed size packets called cells. As cells belonging to different connections traverse the network, they may share network resources such as communication links and ATM switches. It is the task of ATM switches to multiplex cells onto shared link. An ATM switch consists of input port controllers, switching fabric, and output port controllers. Note that delays suffered by a cell at an input port controller and the switch fabric are constant. That is, these delays are same for all the cells. However, the delay at output port controller is different. Cells from different connections are multiplexed there. The delay of a cell depends on the queue length at the output port and the scheduling mechanism utilized. The scheduling policy at an output port controller of an ATM switch determines the order of the cells (from different connections) being transmitted. Typical scheduling policies utilized in the ATM switches are either FIFO or priority driven. However, there are other scheduling policies such as EDF or GPS. Scheduling policies can be classified as either work-conserving or non work-conserving. A work-conserving policy never let the output link go idle if there are cells queued in the output queue, whereas a non work-conserving policy may let the link go idle even if the output queue is non-empty. All work-conserving policies have identical average cell delay, average and maximum buffer need. A non work-conserving policy must by necessity have higher average cell delay, average and maxinum buffer need than a work-conserving policy. On the other hand, work-conserving policies tends to increases the burstiness of the traffic, while non work-conserving policies can be used to limit burstiness.

Data traffic services on ATM will be using the available bit rate(ABR) type of service defined by the ATM Forum. Network bandwidth left over after handling the real-time traffic such as the constant bit rate (CBR) and The variable bit rate (VBR) will be used by ABR or UBR (unspecified bit rate) types of service. End users connected by ABR services will be guaranteed a minimum cell rate (MCR), a peak cell rate (PCR) and a cell loss rate (CLR) that are specified during the establishment of the connection. An ABR source may transmit at a bit rate range between MCR and PCR under the condition that the negotiated CLR is satisfied. However, an ABR type of connection should be delayable and flow controllable. An ABR source may be forced to adjust its transmission rate according to the control signals sent from the network. It is recommended that the ABR service is good for high-performance, non-real-time data applications such as hypermedia document access, file transfer, image transfer, etc. UBR services are often referred to as the best-effort type of services and they require no guaranteed transmission rate, for example, the connectionless datagram services like email. Spare bandwidth left over after processing QoS guaranteed services will be used by UBR under the condition that the service quality of the throughput-or-guaranteed flow is not violated. The CBR and VBR services guarantee the negotiated throughput, the maximum cell delay, and variance. To enable the ABR service to function effectively, a suitable closed-loop flow control mechanism must be implemented. To that end ATM Forum proposed two schemes: (1) rate-based, and (2) credit-based. With the rate-based scheme the controls the transmission rate of the sources to maximize the network performance. Thus, at times, when resources are plentiful the network will allow a source to increase its rate of transmission, but at other times when the traffic is heavy the source rate will be throttled to a safe value. With the credit-based scheme, the network regulates the flow of traffic by sending credit tokens to sources. The frequency of transmitted tokens and the amount of credit are related to the availability of the resources at the entry points to the network, in a manner designed to optimize network performance.

3. CONGESTION CONTROL

Congestion [2][4][6] is defined as a condition that exists at the ATM layer in the network element such as switches, transmission links or the cross connects where the network is not able to meet a stated and negotiated performance objective. The term congestion is often used to refer to a phenomenon whereby a higher offered load to the network leads to lower throughput of a link in order to achieve lower delay. Congestion is a rather complex combination of symptoms. Among these symptoms are higher delays, increased buffer overflow events, throughput decrease, and high call rejection probability. Possible congestion causes are heavy traffic, network mismanagement, poor routing and flow control decisions, performance degradation in certain switches, link failures, various kinds of deadlock, etc. The effectiveness of congestion control mechanisms is not evaluated by means of some congestion metric. It is rather measured in terms of Quality of Service (QoS) supported by the network. Obviously, congestion control may imply any preventive method designed to protect the network from being congested. However, the usual meaning of congestion control is much narrower. Routing, good topological design, network reconfiguration, etc. may constitute effective preventive measures against congestion, but they are not considered as congestion control measures. Congestion control usually refers to methods of controlling the information flow inside the network and at its input. Thus, Traffic control defined a set of actions taken by the network to avoid congestion. Traffic controls measure to adapt to unpredictable fluctuations in traffic flows and other problems within the network. Traffic control mechanics allot the limit resource according to the QoS of services. Traffic and congestion controls operate at three levels:

1. at network level, to perform all functions relating to connection admission control and capacity assignment per link.
2. at call level, to perform resource allocation which maintains a balance between an acceptable QoS and network utilization by limiting the number of connection calls in the network. Here the offered traffic can be characterized by its peak cell rate, average cell rate, burstiness, etc.
3. at cell level, the misbehaving traffic cells can be delayed or discarded. At this level congestion control can be implemented using means such as traffic policing (e.g. leaky bucket or virtual scheduling algorithm), or traffic shaping (e.g. rate or window control).

Two types of control mechanisms have been used to control congestion: window control and rate control. Window control is the more common technique; rate control has also been used . Window control mechanisms have been used to control congestion since the inception of packet switching in the 1960s. A simple model of sliding window control for a single virtual circuit is end-to-end, from source to destination. The sliding window control is one in which each packet is individually acknowledged, allowing the window to move forward or slide by once the acknowledgment arrives back at the source. One question is that window adjustment needs to either avoid congestion or to reduce it if deemed. One possibility is to invoke congestion control in phases, reducing the window size gradually if “mild congestion” is detected; then slamming the window shut or reducing it abruptly, to a minimum value, if “severe congestion” is experienced. “mild” or “severe” congestion could correspond, respectively , to different queuing thresholds at a nodal buffer. To avoid having

two thresholds to consider, involving measures of "mild" and "severe" congestion, more recent congestion control mechanisms have focused on exponential reductions of window size when congestion is to be either avoided or controlled when it occurs.

Rate control is a second method used for congestion control and is the method implicitly proposed for the feedback control. Usually the most effective way to counteract congestion is reduce the traffic at the entry points to the network to the sources of information. The general idea is to adjust traffic flow into the network in such a way as to optimize network performance.

The main purpose of common traffic control is to allot resource and reject excessive connection request. ATM traffic control mechanic assigns the resource to the request connection after it was accepted. The resource can't be shared till the connection is deleted. If there is no resource left, new connect request will be rejected no matter how important it is. It has been frequently claimed that the priority scheduling approach has superior "stability" compared with other approaches, because "essential" processes can be assigned high priorities in order to ensure that they meet their deadline. The main function of an ATM switch is routing. Considering the arrival of upsurge services, cells from different in ports need to be exchanged to the same out port at the same time. There will happen a competition. If the congestion happens in the internal of a switch, we call that internal congestion. Since the transport speed of ATM networks is very high, the congestion tends to be deterioration and the network is forced to paralyze. It's a big problem for vendors to think of it.

Usually the most effective way to counteract congestion is to mainly base on preventive approaches such as call admission control algorithm and reduces the traffic at the entry points to the network. This can be achieved by feedback signaling from the network to the sources of information. The general idea is to adjust traffic flow into the network in such a way as to optimize network performance. A number of rate-based control schemes have been proposed and good reviews are given in ref's [1] [2][3][4]. Yet the ATM equipment providers aim at the network utilization and the increasing demand of large capacity. In a commercial network where pricing is directly related to bandwidth reservation, such a low utilization is highly undesirable. There is no way to shun the congestion control problem. Fig.1 shows the procedure method of congestion control. When more than one cell compete for the same out-port. The congestion arbiter will queue the cells according to the priority and time limit of each cell. Then the most important cell will be transmitted and the least important cell could be discarded.

3.1 Call Admission Control

The objective here is clearly very simple: given a call arriving, requiring a virtual connection with specified QoS (bandwidth, loss probability, delay, etc.), should it be admitted? The role of any Call Admission Control (CAC) [18] scheme is to ensure that the admittance of a new flow into a resource constrained network does not violate the service commitments made by the network to admitted flows. The service commitments may be in quantitative terms of guaranteed cell lose or delay since cell delay can often be controlled within a desired bound by adjusting the buffer size, the traffic can be characterized in terms of cell loss. When a flow requests real-time service, the traffic must be characterized in a such a way that the network can make its admission control decisions. However, it is difficult to characterize the traffic because of the burstiness in the traffic and the uncertainty in traffic parameters. There have been many proposals about admission control and traditional real-time services will usually use peak rate assignment due to the difficulty in anticipating characteristics such as an average rate or a mean burst length. Typically, approaches to CAC like those used for guaranteed service make a priori characterization of sources to calculate the worst-case behavior of all existing flows in addition to the incoming one. A second approach uses a two state Markovian model, and computes the probability of the instantaneous aggregate rate exceeding the available capacity. While this method is more efficient than first one, it fails to take advantage of the buffers. These approaches suffer from several drawbacks. First, it is usually difficult for the user to tightly characterize his traffic in advance. Second, though these approaches give a hard or absolute bound on the cell loss or the delay, they will lead to inefficient network utilization. Third, there exists a modeling tradeoff between the ability to police and the statistical Multiplexing gain. Stochastic models such as those based on effective bandwidth are better suited to achieve good statistical multiplexing gain, but at the expense of policing.

3.2 Forward Explicit Congestion notification (FECN)

FECN [6] [13] is an end to end flow control scheme which uses a binary feedback mechanism. Each data cell transmitted by the source sets a forward explicit congestion notification bit, may be in the Payload Type Identifier- (PTI)field in the ATM header to indicate that no congestion has been encountered. The main advantage of this scheme is its simplicity: a single status bit in data cell is used to indicate a congestion and a single control bit in the RM cell is used to inform the source. This scheme has some undesired behaviors. The source transmission rate increases slowly to the maximum rate leading to a low system utilization. Moreover, the congested switch will mark all input VCs regardless of uncongested VCs. This will create 'beatdown" problem for a fair share allocation.

3.3 Backward Explicit Congestion Notification (BECN)

With BECN [6] [14], the switch estimates the traffic load by monitoring the queue length. When the queue length exceeds a predefined specific threshold, the switch sends a congestion notification cell directly to the source for each virtual connection. By transmitting the RM cells directly to the source the response time is significantly improved compared to the FECN scheme, on account of the shorter communication path length. The downside of the BECN scheme is the increased hardware cost of the switch needed to monitor the congestion status of the switch and to manage transmission of the RM cells.

3.4 Explicit Rate (ER) Feedback Notification Schemes

Instead of using only the proportional increment and decrement of the transmission rate algorithm based on FECN scheme, the explicit rate feedback notification is adopted for the Explicit Rate (ER)[6] [15] scheme. In this scheme, the RM cell travels forward through the network, it is now called Forward RM cell, providing the switches with the information necessary to determine the correct bandwidth allocation among ABR connections. The explicit rate indicator may be marked in RM cells by the switches. Switches supporting the EFCN mechanism will only ignore the content of the RM cell. Howver, the overall performance is not degraded when ER-based switches are inter-operable with EFCI based switches.

3.5 Explicit Rate with Congestion Avoidance Schemes

Instead of reacting to congestion, the OSU scheme makes use of congestion avoidance[6][16]. The OSU scheme also uses

explicit rate indication but does not explicitly require the source's desire rate; the source uses a rate adjustment factor instead. The scheme works as follows: the source monitors its load and periodically sends control cells that contain the load information. Instead of using queue threshold detection, each switch periodically computes its input load compared with target utilization. The switch target utilization is determined with a margin of safety required to avoid a congestion.

3.6 Segmentation of Rate-Based Flow Control

To reduce further end-to-end delay in the control loop, a network in the rate-based system can be sub-divided into a number of sub-networks[6]. The intermediate switches at entry and exit points to a sub-network act as a Virtual Source (VS) and Virtual Destination (VD), respectively. In this way, the delay in the feedback loops is individually reduced leading to faster response to changes in the traffic intensity and, consequently, utilizes resources at the bottleneck better.

3.7 Start-Stop Flow Control

Start-Stop Flow Control[6] [17] uses hop-by-hop per VC congestion flow control. The memory pool in the switch consists of two parts: (1) private buffer area, and (2) shared buffer area. In the private buffer area space is reserved for each active VC. This is adequate to provide for minimum throughput, thus avoiding deadlocks. The remainder of buffer resources is held in the shared buffer area and is available to all active connections and is assigned by the switch dynamically as the need arises. In this way, efficient statistical multiplexing can be achieved. Flow control is enforced by two modes, on and off: data cells are transmitted during the on period and no transmission is allowed during the off mode. The onset of the on and off modes is generated by the switch depending on the queue length in the buffer. When the queue length reaches the specified threshold, the stop signal is activated.

4. MODEL BASED ON MULTI-MARKOV

In this paper, we adopt priority control algorithm for providing different QoS bearer services can be implemented by using threshold methods at the ATM switching nodes. In the separate route approach, buffer management is not required at the switching nodes, since priority processing is executed at the connection level by the routing function. Priority disciplines in queuing theory can be categorized into two major types: 1). Service or delay priority disciplines, which govern the time at which cells in the buffer are transmitted. 2) Buffer access or space priority disciplines, which govern the input access, or space priority disciplines, which govern the input access of cells into the buffer. ATM network traffic control model is usually built according to the type and usage of services. In general, the model of non real-time services is based on Poisson Model. Real time services and data flow is built as MMPP Method or Multi MMPP method. Considering the cells of the same connection can have different priorities but same demand of discard/accept sequence. The vector C_i (t_d, t_0, p_i) denotes the request priority of cell i, t_0 indicates the start time, t_d means the time deadline, so $|t_d - t_0|$ means time left for a cell to be sent; we take p_i as the priority of this cell. So the time limit and the priority can be considered at the same time.

For the input queued cells, we use a threshold buffer with finite size K (cells) for each outgoing route to switching matrix. The queuing analysis can be focused on a buffer associated with any particular outgoing link. Assume there lays a discrete-time slotted queuing system in which each slot duration is equal to an ATM cell transmission time and is assumed to be of unit length. Cells start transmission only at the start time of a slot. Let the ratio that total cells of an input line to an output line be Y_{ij} there will be:

$$\sum_{j=0}^{N} Yij = 1, N = total\ ..cells$$

Let p_i be the probability of the cell i in the buffer at the end of a time slot. Let $\prod_{ij}$ denote the transmission probability of a cell to be transmitted from the input line i to the output j. As for the total cells, we get the following formula. So the priority and deadline of each cell are considered at the same time. The ATM arbiter could classify the cells according the QoS of each connection.

$$\Pi = \begin{Bmatrix} C_0 & C_1 & \dots & C_n & 1-\sum_{i=0}^{K} C_i \\ & Y_{11}C_0 & \dots & Y_{11}C_{n-1} & 1-\sum_{i=0}^{K-2} \frac{p_i}{|t_d - t_0|} C_i \\ & & & \dots & \\ & & & Y_{1n-1}C_1 & 1-\sum_{i=0}^{K-2} \frac{p_i^{K-1}}{|t_d - t_0|} C_i \\ & & & & 1-\sum_{i=0}^{K-2} \frac{p_i^{K}}{|t_d - t_0|} C_i \end{Bmatrix}$$

$$Y_{ij} = \frac{p_i^i}{|t_d - t_0|},\ i=0, 1, \dots, K$$

For example, if there are 2 different kinds of service compete for an output line, one service has high priority while the other has low priority but limit time deadline. Thus the arbiter will compute the possible cell loss rate and decide to hold the low priority cell or the high priority cell.

According to the loss probability of each cell, all the cells could be stamped and dealt well. From the probability matrix, the condition of all the cells can be divided clearly. So we can transmit the urgent cells, add time-delay to the important cells so that they could be transmitted later. If necessary, some less important cells are discarded to avoid congestion. With this mechanic, the arbiter work well.

5. EXPERIMENTS AND CONCLUSION

We build the experiment environment as follows: Different real-time services such as voice, video, real-time compression data or fax are adopted by the Input Control line i and exchanged to the Output Control line j. After clock recovery and SAR (segment and resemble), they are sent to an ATM Switch to exchange to the correct route. To emulate the congestion circumrotation, we set the head of those service cells in the same VPI/VCI. We use cell loss detection machine at the Output Control line to check the cell loss rate. Fig.3 shows the buffers size and the cell loss rate. From Fig3 we know that the cell loss rate is linked to the buffer size. When we queue the cells according to the model discussed in Section 4, the cell loss rate improves greatly. This method proved to be effective in cell loss at the cost of a tolerant degradation of the behavior of the network.

This paper proposes a theoretical analysis of congestion control mechanic. The cell loss probability performance of the

buffer threshold scheme to implement the priority assignment arbiter is also discussed. According to this algorithm, we can order all the cells that waiting for transmission. Thus cells of different deadline and priority could be deal with according to the necessity. The ATM network could offer the best QoS. Experiments show that his mechanic is effective in ATM congestion control. Future work will concentrate on UPC control and CAC algorithm to assign the traffic of each channel.

6. ACKNOWLEDGMENTS

This work is partly supported by the National Natural Science Fund, Natural science Fund of Guangdong province, "Thousand, Hundred, and ten" outstanding person fund of Education Department of Guangdong Province, Natural science fund of Education Department of Guangdong Province.

7. REFERENCES

[1] Ermedahl, et al., Response time guarantees in ATM networks, Proceedings of the 18th IEEE Real-Time Systems Symposium, 274-284, IEEE Computer Society Press, 1997.
[2] A.E.Eckberg, et Al. Meeting the Challenge Congestion and Flow Control Strategies for Broad Band Information Transport. GLOBERCOM'89, 1989, pp 1769-1773
[3] J.K.Ng et al., Integrated delay analysis of regulated ATM Switch, Proceedings of the 18th IEEE Real-Time Systems Symposium, 285-296, IEEE Computer Society Press, 1997.
[4] Saunders. ATM Forum Ponders Congestion Control Options, Data Communications, 1994, Mar, pp 55-60.
[5] Chung G.Kang, Harry H.Tan, Queueing analysis of explicit priority assignment buffer access scheme for ATM networks, Computer Communications21 (1998), ELSEVIER, pp 996-1009
[6] S. Kamolphiwong et al., Flow control in ATM networks: a survey, Computer Communication 21 (1998) 951-968.
[7] W.Fischer et al., Data communications using ATM: architectures, protocols, and resource management, IEEE Communications Magazine, August, 1994, 24-33.
[8] F.Bonomi, K.W.Fendick, The rate-based flow control framework for the available bit rate ATM service, IEEE Network Magazine, March, 1995, 25-39.
[9] C.S.Wu, Link-sharing method for ABR/UBR services in ATM Networks, Computer Communications 21, 1998, 1131-1142.
[10] A.Song et al., Efficient delay computation methods for an ATM network with real-time Video Traffic, Proceedings of the 20th IEEE Real-Time Systems Symposium, IEEE Computer Society Press, 1999.
[11] C. Li, Static priority scheduling for ATM networks, Proceedings of the 18th IEEE Real-Time Systems Symposium, 264-273, IEEE Computer Society Press, 1997.
[12] ATM Forum, Traffic management specification, Version 4.0, 1995.
[13] B.A. Makrucki, explicit forward congestion notification in ATM network, proceedings of TriComm'92, February, 1992.
[14] P.Newman, Backward explicit congestion notification for ATM local area networks, IEEE GLOBECOM, 1993, 719-723.
[15] A. Charny et al., Congestion control with explicit rate indication, ATM Forum/94-0692, July, 1994.
[16] R.Jan et al, The OSU scheme for congestion avoidance in ATM networks using explicit rate indication, Proceedings of First Workshop on ATM traffic management, Paris, 1995.
[17] J.Murphy, Bandwidth allocation by prcing in ATM networks, IFIP: Broadband Communications, 1994, 333-351.
[18] S. Lee and J.Song, A measurement-based admission control algorithm using variable-sized window in ATM networks, Proceedings of ICICS'97, IEEE Press, , 378-384.

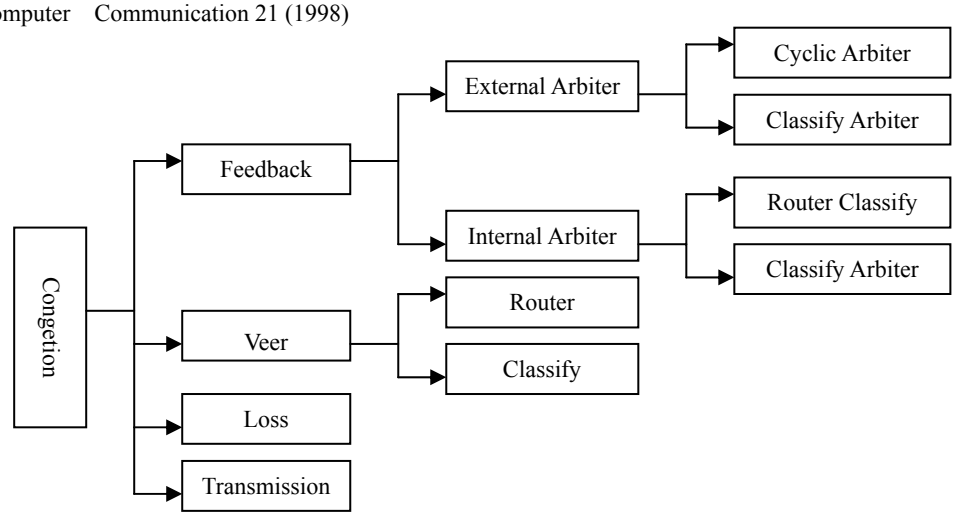
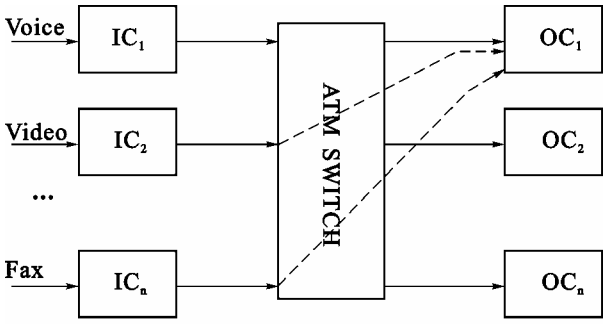
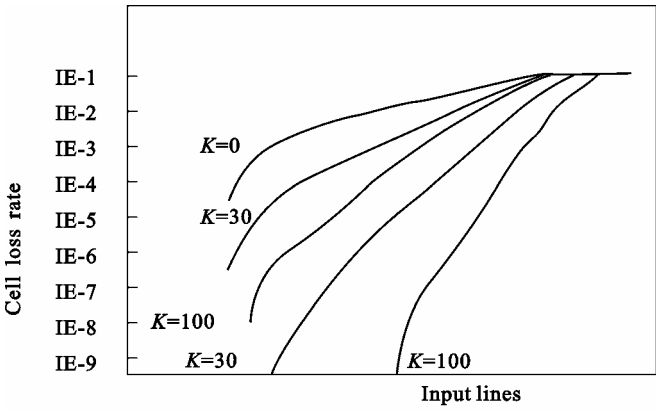


Fig.1 ATM congestion control procedure



IC: Input control line I OC: Output control line

Fig. 2 Experiment and Device



K: Buffer size *K*: After queue

Fig 3 Cell Loss Rate and Buffer Size

Computer intellective net research base on the ANN

Shengjun Xue
Institute Computer science and Technology,
Wuhan University of Technology
Wuhan, Hubei, 430063, China
E-mail: sjxue@mail.whut.edu.cn
And
Ran Tan
Institute Computer science and Technology,
Wuhan University of Technology
Wuhan, Hubei, 430063, China
E-mail: tanran@public.wh.hb.cn
And
Jing Liu
Institute Computer science and Technology,
Wuhan University of Technology
Wuhan, Hubei, 430063, China

ABSTRACT

The paper introduces to make use of ANN connatural abilities which is self-learning, self-organize and self-classify, and to adopt parallel processing technique, which can realize the zero latency time of every processing cell communicates each other in intelligence communication net and the checking and disposal of communication signal, and it can realize net reconfiguration automatically.
Keywords: ANN; Intellective net; Knowledge base; Distributed

1. INTRODUCTION

Artificial neural network (ANN) is a kind of model which simulates work machine-made of human brain, because it is a kind of reticulation structure which is made up of many same or alike nerve cells by broad connect, and adopts the distributed parallel processing algorithm. Compare to traditional processing way, it has a lot of advantage. First, ANN adopts distributing storage means to express information, the information, such as different kinds of signals, communication protocols and rules and so on, which are needed by net communications, are contained impliedly in the nerve cell and the joint authority of ANN. This is constitutionally different from the traditional expression methods; Second, ANN deals with the information by the parallel processing method, it is different form the traditional method which matches and propels rules an item by an item, it allows to dispose plenty of information at the same time; Otherwise, ANN can accomplish complicated unlinear mapping, this is a kind of self-adapted learning process, it is put up the ability of abstract thinking. In a way, it is similar to thinking mechanism of human brain. ANN has some abilities, such as study, memory, association think, fault tolerance, parallel process and so on. So we can assume to make use of ANN connatural abilities which is self-learning, self-organize and classify, and to adopt parallel processing technique, which can realize the zero latency time of every processing cell communicates each other in intelligence communication net and the checking and processing of communication signal, and it can realize net reconfiguration automatically. It has a wonderful application foreground.

Therefore, this article discusses firstly the push inference mechanism of neural network knowledge base. Using neural network model and by offline training, the push inference mechanism in the neural network knowledge base is stored impliedly in the joint authority of neural network. By online learning and the adjustment of control authority, hoping to reach the anticipant output control.

2. NEURAL NETWORK KNOWLEDGE BASE

Neural network knowledge base is made up of many information cells. Each information cell describes an attribute value, concept value or neural network block. Each neural network block is equal to multiterm of rules of traditional knowledge base; the connection adopts the form, which is Back-Propation configuration chain. When some information cells are activated, some other neural network block will be activated, thereby, leading to the lower lever information cells are activated. This process will go on spread until no new information cell is activated. This process forms a kind of push inference decision net. If the information cell that is activated at the first time only has one, it will be regarded as the root node. This can form a decision tree (if there are N information cells which are activated for the first time, each information cell will be as a root note, then can form N decision trees). Inference mechanism in ANN knowledge base is made up of two parts, one is the positive inference the other is the reverse inference. The positive inference process: first, according to the given condition, it will run the width priority search. Then according to the condition of the information cell in neural network knowledge base is activated, it will produce automatically conclusion or new information cells. Both the produce of each new information cell and the new information cell join neural network knowledge base should ensure the no redundancy, consistency and compatibility; Reverse inference mechanism will decide automatically that which information cell in the neural network knowledge base push-up this original condition. At this time, this original condition can be explained an enquiry, and the action of the ANN reverse inference can be shown that trying to use the coded information to answer this enquiry.

Except positive inference and reverse inference, neural network knowledge base should include auto-maintenance mechanism. That is to say, when the obtain system of never net information makes the obtained information cell add to its knowledge base or execute the operation of removing information cell, ANN should be able to check, update and

modify the information in the knowledge base automatically, removing redundancy information, and find contradiction and can recover the information during the inference process.

3. NEURAL NETWORK MODEL AND ALGORITHM

Supposing the short cutting fuzzy inference is:

$$U = \frac{\oplus_j^m - o a_i B_i}{\oplus_i^m - o a^i} \qquad (1)$$

$$a_i = \otimes_j^m - o \mu A_{ij}(X_j) \qquad (2)$$

In which, R_i express the number i item control rule; X_i and U express the import variable and the control variable respectively; B_i is the verdict of rule; A_{ij} is the fuzzy subset that is defined by the fuzzy subjection function $\mu A_{ij}(X_j)$; "$\oplus$" express algebra sum and "$\otimes$" express algebra product.

Using proper node function links with authority, then there will be a multiplayer topological structural neural network that contain the hidden layers. The middle hidden layer finishes the fuzzy composite algorithm. The number of this layer's nodes is equal to the items of inference rules. The front layer finishes subjection function, which can realize fuzzy. The back layer finishes the judgment function of fuzzy. It can be realized by the single layer notes or by the net that is formed by the multiplayer notes.

In order to make the input-output reach the anticipant mapping, the two net authority value W_{jk},$W_{i,j}$ are adjusted by net training, which will make error function E go to least.

The authority value amended formulas are:

$$W_{ij}(t+1) = W_{ij}(t) - \eta(t)\frac{\partial E}{\partial W_{ij}} + \beta[W_{ij}(t) - W_{ij}(t-1)] \qquad (3)$$

$$W_{jk}(t+1) = W_{jk}(t) - \eta(t)\frac{\partial E}{\partial W_{jk}} + \beta[W_{jk}(t) - W_{jk}(t-1)] \qquad (4)$$

Error function is:

$$E = \sum_k \sum_{i=1}^m (d_{ik} - 0_{3ik})^2 \qquad (5)$$

In which, η (t) is self accommodation study ratio, β is momentum factor. 0_{3i} is the nubmerIII layer nodes actual output of the fuzzy neural network.

$$\frac{\partial E}{\partial W_{ij}} = \delta_i \cdot O_{2j} \qquad (6)$$

$$\frac{\partial E}{\partial W_{jk}} = \delta_i \cdot x_k \qquad (7)$$

$$\delta_i = O_{3i}(1 - O_{3i})(d_i - O_{3i}) \qquad (8)$$

$$\delta_j = O_{2i}(1 - O_{2i}) \cdot \sum_i \delta_i \cdot W_{ij} \qquad (9)$$

Self-accommodation study ratio is:

$$\eta(t+1) = C \bullet \eta(t) \qquad (10)$$

C is the study factor, it chooses (0.7~1.0) which makes the speed of study faster.

The process like about can be finished by offline learning. Taking out the tagged of net input is a very important job. The chosen tagged is fit or not, which will decide straightly the result of the final discrimination good or bad. Commonly, the choose of tagged has some principles: First, it is sufficient principle, that is to say the choose of tagged should keep the information of former model fully; Second, on the base of satisfied sufficiency, it should try to reduce the tagged dimensions; At last, it is convenience principle, it is to say the amount of calculation that is spent on choosing tagged can not too large because it will affect the speed of system's discrimination.

4. SELF-ACCOMMODATION NERVE CONTROL

In the Self-accommodation nerve control process, its self-accommodation parameter is the authority coefficient of neural network. By proper study the object of unknown character can be controlled and can accommodate the environmental change. The control scheme of this article is:

Define the error function:

$$E^\infty = \frac{1}{2}\sum_{i=1}^m (U_{di} - U_i^\infty)^2 \qquad (11)$$

In that function, U_{di} is expectation input, U_i^∞ is the actual output of net.

The authority value revision formulas are:

$$W_{oi}(t+1) = W_{oi}(t) - \eta(t)\frac{\partial E^\infty}{\partial W_{oi}} + \beta\Delta W_{oi}(t) \qquad (12)$$

$$W_{ij}(t+1) = W_{ij}(t) - \eta(t)\frac{\partial E^\infty}{\partial W_{ij}} + \beta\Delta W_{ij}(t) \qquad (13)$$

$$W_{jk}(t+1) = W_{jk}(t) - \eta(t)\frac{\partial E^\infty}{\partial W_{jk}} + \beta\Delta W_{jk}(t) \qquad (14)$$

By online learning, adjusting authority value coefficient can make the deviation reach the least, and make the output adjust to the hoping output control value self-accommodatingly.

5. CONCLUSION

The process control of self-organized never net is: giving the beginning authority value firstly, the beginning rate of learning and characteristic subclass, by the training of the algorithm of learn, we can put the all of ratiocination, maintenance and the decision making which included in the knowledge base of never net to fulfil to the never net's three layer authority value which is $W_{ij}(t)$, $W_{jk}(t)$. So the relation or the rule of input and output can concealed memory in the joint authority value of net, the basis to adjust authority value, At the same time to put the inaccuracy of output unit to input unit conversely layer by layer, so each layer unit may get " a share proportion", by above way we can get each layer's reference inaccuracy to adjust relevant joint right, and to make this kind of inaccuracy to a satisfied level up to.

The never net composed by the above-mentioned model and the knowledge base of the never net which has the below feature: the distributed-memory technology is adopted, the principle to memory information almost as same as the human brain which means it has the ability to deal wholesale parallel processing and self-adapted learning capacity, it can not only to deal with the concentrate knowledge but also to deal with the new knowledge especially be skilled at those problems which is hard to describe by algorithm but is easy to learn by lots of examples, all of above is very important to automatic identification system of parallel processing and the function of automatically organize net which is concerned by computer intellect net.

However, we should notice, as a result of the complexity of the intellectual net's structure and the limitations of the never net's scale, we have lots of actual difficulties to conform a whole intellectual net system, the way to keep on to improve its functions is to conform a even more large knowledge base of never net, by this way we could perfect the ability of automatic dealing of the intellectual net

.

6. REFERENCES

[1] Xue Shengjun Gao Xiaohong Xiong Qianxing. Parallel Design Model and Parallel Design Method Based on Neural Network IEEE ICIP'97

[2] Xue Shengjun,, Tan Ran, Shang Lei. Intelligent Communications Networks Fuzzy Control Technique ICAIE'98

[3] Hin-ichi Horkawa et al. On Fuzzy Modling Using Fuzzy Neural Network With the Back-Propation Algorithm. IEEE Trans Neural Networks, 1992,3(5) .

[4] Yashiaki I et al. Neural Network Application for Direct Feedback Controllers IEEE Trans Neural Networks,1992,3(2)

[5] Psaltis D et al. A Multi-layered Neural Network Controller, IEEE, CSM, 1988,4(17)

[6] Ichiro Enbutsu et al, Fuzzy Rule Extraction form a Multilayered Neural Network Proc.IJCNN'91

[7] Aiken S W, A Parallel Neural Network Simulator IJCNN'90

Technology Of Intelligent Meta-search Engine Applied In Network Information Value-added Service

Li Liu
College of Information Technology, Southern Yangtze University
WuXi, JiangSu ,214000, China
E-mail: wxliuli@sytu.edu.cn

And

Wenbo Xu
College of Information Technology, Southern Yangtze University
WuXi, JiangSu ,214000, China
E-mail: xwb@sytu.edu.cn

ABSTRACT

After having analyzed the development of existing Network Information Value-added Service, the model of Intelligent Meta-search Engine was proposed and applied to a business case. The design and relativity of information acquisition, extraction, integration and communication were given in details.

Keywords: Network Information Value-added Service, Distribution, Intelligent Meta-search Engine.

1. INTRODUCTION

With the development of Internet, network information service springs up. How to offer information value-added service in the Internet has become a hot issue. ASP(Application Service Provider) is a new approach to provide small and mid-scale companies with professional, suitable and personal service according to the certain career and domain. It is showed that ASP in China is go on the steps, such as China Netcom preparing ASP developing strategy, *Yongyou* corporation, one of the biggest financial software company, attempting to become the application service provider of enterprise financial management systems, *Han Pu* corporation, one of the biggest management consulting company, making fund to enhance application service in ERP (Enterprise Resources Plan) field.

As what you can imagine, if the technology of Information search engine is integrated in Application Service, it will show great power of itself.

Meta-Search Engine is the search engine based on other independent search engines that search information by themselves. It receives user's information search inquisition, format them into internal norm and submit then to many independent search engines that chosen in advance. Ultimately, the results from these independent search engines will be analyzed and integrated into uniform information and send back to the user. So it integrates advantages of differently independent search engines and simplifies search work for users to find out needed information from one search engine to another.

To inherit the advantages of each independent search engine, which is concerned, it is essential to record it according to its discrepancy and characteristics firstly. Then result data are classified and analyzed by the technology of Data Ming. Finally, with user's information search options such as area, topic, real time network condition, etc. the model of meta-search engine schedule strategy is created. Depending on this model, we can submit information search requisition to independent search engines that are dynamically chosen as the suitable engines according to real-time factors, rather than all the same chosen schedule. Obviously, guided by meta-search engine schedule strategy model, the quality of information search service would make great progress.

Since we can locate to higher-relativity web page associated with what users want to acquire directly, the main information can be extracted from them. It is better than just to send back URL address to user end. Information of personalization and diversity can be offered to Network Information Value-added Service.

2. THE TECHNOLOGY OF INTELLIGENT META-SEARCH ENGINE

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.

Meta-search 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. Meta-search engine are simply constructed, just as Meta-crawler.

In the intelligent meta-search engine model [1], each independent search engine servers as a node of the information search service system. Its task is to perform users' information search inquisition. How to allocate the task to these search engines depends on analyzed historical record of each search engine.

Theoretically, search engine knowledge base containing complete information about every search engine is hard to realize, but we can resort to narrow the range of relative search engines, according to specific search theme demand. The knowledge base is a kind of approximate, a kind of local realization of the information search service for this field. According to the certain a field, constructing a local knowledge base is relatively easy. It can't compare with ideal knowledge base on quantity and quality, but is practical for a specific field searching task. More importantly, the knowledge in base can be added, improved and updated.

Intelligent meta-search engine system is divided into four main parts, user interface manager, search engine manager, search agent and Data-Ming manager.Fig.1 illustrates the structure model of the system.

User Interface Manager

User interface manager is responsible for receiving user inquisition. It interprets user operations with system's interface

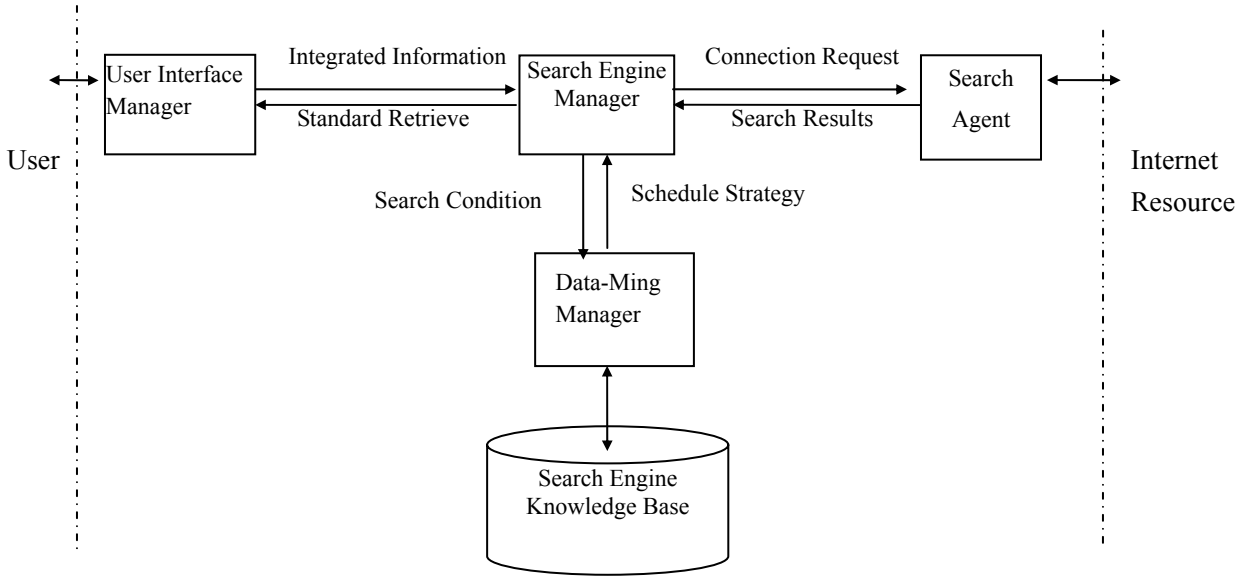


Fig.1 structure model of the intelligent meta-search engine system

into standard systematic inquiry. After search engine manager integrates relative information, it will take the responsibility for exporting search results.

Search Engine Manager
When search engine manager receives standard systematic inquiry. sent by user interface manager, it will send search condition to Data-Ming manager and receive search engine agents schedule strategy. According to the dynamic strategy, connection requests are generated. Then each search engine agent can exert his power to seek relative information. At the same time, it providers with much more functions, such as buffer management and information integrity.

Search Engine Agent
As a browser, each search engine agent is an independent process communicating with a search engine, which is the really part to get information resource on the Internet. So long as we send applications according to HTTP protocol to a Web server, the server will give corresponding reply.

Data-Ming Manager
Data-Ming manager is the main model to embody intelligence of the system. It implements pre-process function and generates search engine schedule strategy.[2]

3. THE APPLICATION OF DATA MINING TECHNOLOGY

Acquisition Of Information From Web Pages
Search engine agent communicates with search engine as a browser. When the connections to the external search engines being created, to which the system get searched information just like a common browser. So long as we send applications according to HTTP protocol to a Web server, the server will give corresponding reply. There are two classes in Java, which can be used to acquire Internet resources. One is the *URL* class and the other is *URLConnection* class.[3] Both of them provide the method of reading information in bit stream and can judge the type of resource information. It is useful make related decision. *URLConnection* class can provide much more information than *URL* class, such as the length, transacting time, updating span, encoded method and resource title.
For instance, if we store the connection URL address of the specific search engine into *Urlstring* variable, the programming code to create connection link can be written as follows:

```
//connectSE.java
import java.awt.*;
import java.net.*;
import java.io.*;
public class getFile extends Applet
{
    String info;
    public void init()
    {   URL url;
        URLConnection urlc;
        try{
            url = new URL(Urlstring);
            urlc = url.openConnection();
            urlc.connect();
            ……/*To deal with connected data*/;
        }catch(MalformedURLException mfe){
            System.out.println("URL form error!");
        }catch(IOException ioe){
            System.out.println("IO Exception!");
        }
}   }
```

Automatically Extracting Metadata Information
Information sent by a Web search engine is described with HTML (Hypertext Markup Language), whose source code is

unreadable for us to process directly. So it is necessary to filter and analyze feeding back information. The source code of HTML usually can 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 will be shown in your browser, such as IE, or Netscape. The concept of analyzing HTML documents can be described as 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, referred as metadata. It can be preceded as following steps: finding out the start position of metadata, getting controlized info within the angle brackets, setting specific signals. After being constructed in a metadata entry, it can be store into a temp table.

Fig.2 shows the process to filter and construct useful information form a HTML source code. In addition, the code in Java language is formed by Unicode encoded method, in which Chinese word as well as English word is dealt as one character. So we have to pay much more attention to Chinese conversion.

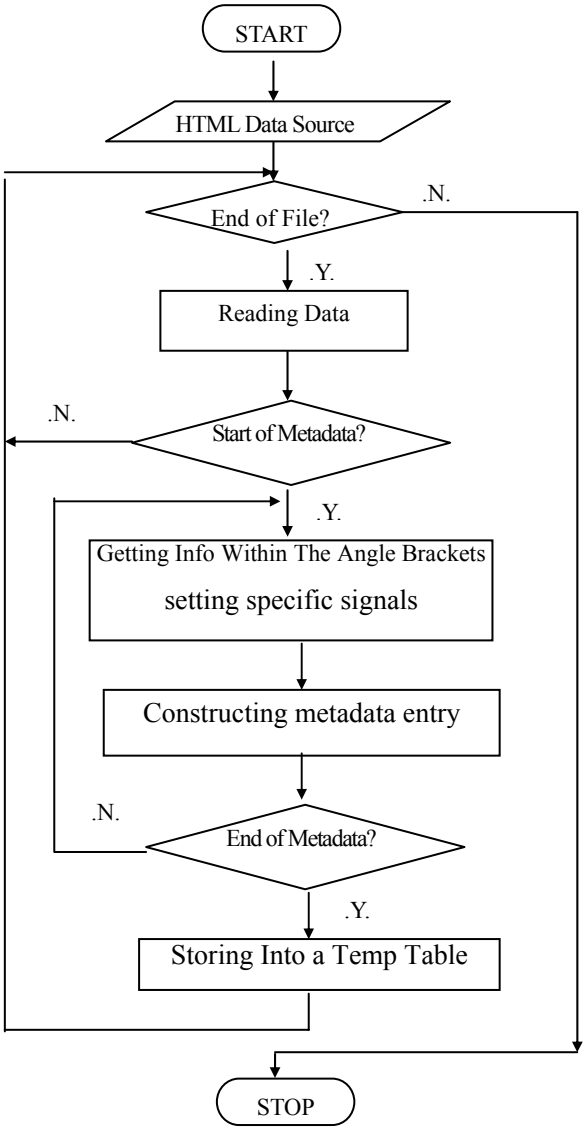


Fig.2 the Process of Extracting Metadata Information

Integrating Information

There is large quantity of information sent back from search engines. It is common to find repetitive information. To improve the performance of the intelligent search engine, it is essential to filter and integrate the information. And it is urgent to enhance the function of search engines in this aspect. There are three main process methods to be proposed.

(1) Same URL Address:

We will choose the newest information record and deal with as metadata.

(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 the information; marking a copy of content with several URL addresses that can be visited.

(3) Metadata with Same Key Information:

In the information service of market, we mainly take care of products information such as product name, manufacturer, product model, current price as well as price tendency. When it is detected multiple information of the same product name and product model with different price, there are much than we can just to arrange them one by one as manufacturer's order. We can analyze the range of the price of products according to season, area or manufacturer. We also can calculate the average merchandise price of each manufacturer. Each of these kind of processing techniques differs from the data quantity, storage space and information presentation. User can select the suitable method as they wishes.

Sending Information with XML

Today, Internet has pervaded into each fields of human lives. XML using as a intermediate data interface has showed its importance in electronic data exchanging. Associated with E-Business, it can be used in information Value-added service to dispatch Information. In systematic interface design, combining with CSS or XSL technologies, XML can be used to realize the various display patterns on Web browser. Middle layer need a agency program to operate, to access data in database system and to output XML document. At the same time, different programming language and script language need different SQL API and XML grammar analysis. Since the Java technology being used in system programming, JDBC and DOM interface are needed to get data from database [4],in order to generate dynamic XML document. Fig.3 shows mechanism of storage and access of XML data. Such combination improves the reliability, portability and flexibility of the network information Value-added service.

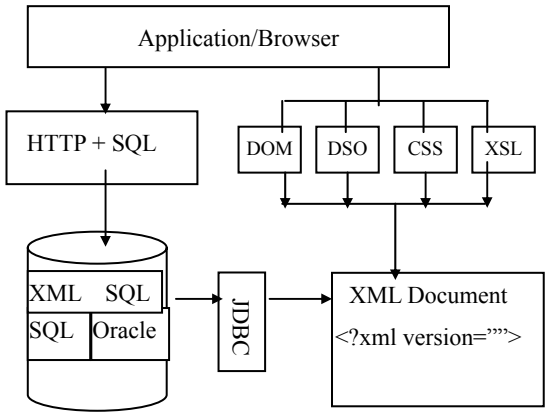


Fig.3 Mechanism Of Storage And Access Of Xml Data

4. CONCLUSION AND RESEARCH PROSPECT

For example, we tried to provide information Value-added service of IT products to Wuxi government bidding system and have reached great progress. In the future, we will make further development of intelligent Meta-Search engine to improve quality of network Value-added information service.

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 Meta-Search engine.

5. REFERENCES

[1] Liu Li, Sun Yan-tang. Internet Information Search Service Based on the Technology of Data Mining. International Symposium on Distributed Computing and Applications to Business, Engineering and Science Proceedings. Hubei Science and Technology Press,2001.213～216.

[2] Jiawei Han, et al. Concept and Technology of Data Mining. Beijing: Mechanical industrial press, 2001.185 ～ 222.

[3] Tong Xu. Visual J++ programming [M]. Renmin postage press, 2000.331 ～ 334.

[4] CharlesF.Goldfarb, Paul Prescod, XML handbook. Renmin postage press, 2000.9 and 341 ～ 364

Automated Network Management with SNMP and Control Theory*

Yijiao Yu, Qin Liu, Liansheng Tan and Debao Xiao
Department of Computer Science, Central China Normal University, Wuhan 430079, PR China.
E-mail: {yjyu, liuqin, l.tan, dbxiao}@ccnu.edu.cn

ABSTRACT

Network management system (NMS) is required not only to have network performance monitoring function but also to have automated real time network control. Most NMSs configure devices manually without processing data from SNMP agents. In order to make it automatic, it's suggested that automated NMS should be built based on SNMP and control theory. This proposed that Manager-Agent network management model is similar to the closed-loop control model. According to device features, local network control systems are divided into two kinds, one is switch control, and the other is server control. The engineering implementations of the two control systems are analyzed in detail. How to choose a sampling time in network control system is discussed because it's difficult in almost every network control system design. We find that sampling time should be close to the agent sampling time of devices in centralized control model, and it should be also larger than time delay of Internet in distributed network control model. To show how to realize network management with control theory, an example about admission control system in a genetic algorithm computing server is illustrated step by step and control effects are tested. The experiments results support the novel approach is efficient to automated network management.

Keywords Network management, Control theory, SNMP, Closed-loop control, Real time.

1. INTRODUCTION

Network configuration management means controlling network status through setting arguments of devices. Network configuration can be regarded as network control process. Most popular Network Management Systems (NMS) can only monitor network performance status, such as collecting data from devices with SNMP, saving them in database and showing them in GUI to network masters. They are only network monitoring, and not network management and control.

Configuration management is completed by hand, such as login remote devices using TELNET, inputting commands and so on. Obviously the manual way is not able to satisfy the requirement of real time and automated network management. The only way to real time in network performance management is to reduce people's activities, code human decisions into program and run them by NMS automatically.

There're a lots of automated control systems in industry, such as automobile steering control system, multi-objects searching and tracking system, robots and so on. Looking into these control systems, it's evident that the basic theory and tools are control theory and computers. For there're lots of computing tasks in controller while computers are powerful computing tools, they are combined in control system.

Since 1990, computer networks become more and more large and complicated. At the same time, network management is becoming more and more challenging. In order to control such complicated networks, control theory was utilized in computer network research. For example, before 1990, most network congestion control algorithms were not described in complicated mathematics model, such as Sliding Window Protocol, Random Early Detection algorithm. However, as time goes on, those algorithms were found that they were insufficient to cope with congestion control very well. Control theory was borrowed to network congestion control [1][2] and had achieved better control effects.

Performance management must be real time and supported by automated configuration management. A novel automated network management solution is proposed in this paper, which is based on SNMP and control theory. The contents of the paper are organized as follows. Network closed-loop control model is in section 2. Section 3 shows the automated network control approaches in engineering. How to select sampling time is discussed in section 4. In section 5, an admission control system on genetic algorithm server is designed to test the novel approach, and experiment results are shown in it. Finally conclusions are drawn.

2. NETWORK CONTROL MODEL

Closed-loop control model and network management

To design an optimal control system, selecting appropriate control model must be the first task. There're two control models, one is open-loop control model, and the other is closed-loop control model. The former requires the controlled objects can be described formally, but the later needn't those conditions. Computer network is so complicated that it's impossible to model it only with some equations, although there're some conclusions about its traffic model, congestion features. In the closed–loop control system, controller adjusts control arguments according to the feedback information, then the controlled objects can work on an optimal status. Because the assumption is not very strict, closed-loop control model is used frequently in applications. The model is displayed as Fig.1.

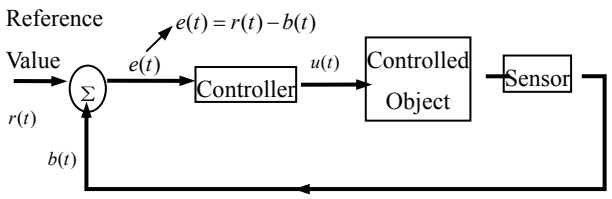
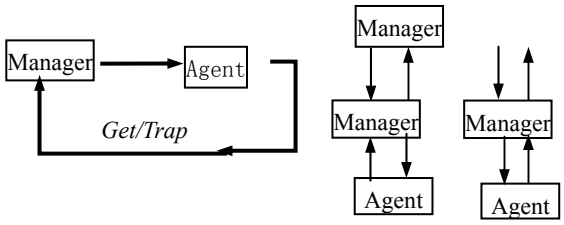


Fig.1 Closed-loop control model

Network management models can also be classified into two categories. One is centralized management model and the other is distributed model. The detailed architecture about two models is illustrated in Fig. 2. Comparing Fig. 1 with Fig. 2 (a), they don't match exactly.

* The research is supported by the grant of Key Science & Technology Foundation in Hubei Province (2001AA104A05) and Natural Science Foundation in Hubei Province (2001AA104A05).



(a) Centralized model (b) Distributed model

Fig.2 Network management models

To realize automated network management, some transforms about network management model should be done.

Centralized network control model

Agent in Fig.2 (a) is a software process in switch, router or server. It monitors statuses of devices and writes records in Management Information Base (MIB). Agent can accept SNMP requests and carry out corresponding operations, such as retrieving data when receiving Get primitive and sending fault traps when device is not normal. MIB data are status information of controlled devices. So we can turn the Fig.2 (a) into Fig.3 without changing the physical essentials. Fig.1 and Fig.3 are similar. Devices in Fig.3 can be regarded as the controlled object in Fig.1 and the agent can be looked as a sensor. One difference is that Fig.3 lacks a reference point in manager.

Fig.3 The centralized network management model

Manager is the core of NMS. In advanced NMS, users are permitted to set an expert value or threshold. For example, if there's a threshold about trap level in trap information filtering, some traps will not be notified when their priory is no more than the threshold. Based on the above discussion, we can see that reference point and reference value exist in manger actually. Fig.3 can be transformed into Fig.4, which is a standard closed-loop control model and possible to do automated network control with closed-loop control approach.

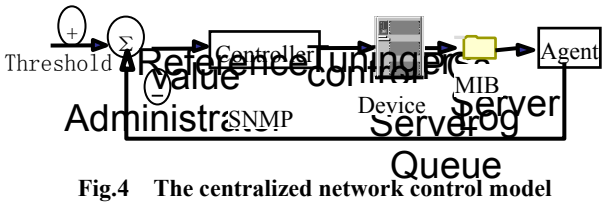


Fig.4 The centralized network control model

Distributed network control model

Distributed network management model can be transformed into Distributed Network Control Model (DNCM) as Fig.5. There're two loops in DNCM, one is the local loop, and the other is remote loop based on Internet.

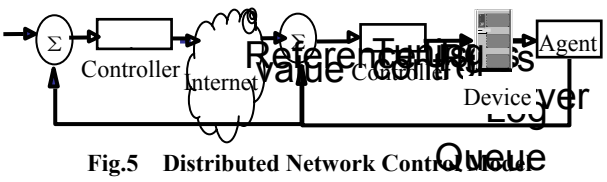


Fig.5 Distributed Network Control Model

The local loop is a centralized network control loop. With small transmission delay and wide bandwidth, it has good real time feature and can configure devices directly with Set primitive in SNMP. With regard to remote loop, the local manager (lower level manager) is a controlled object, so the local manager should support getting commands from remote manager (higher level manager) and do responds. Internet means open and insecure, therefore, local manager should verify whether the remote requests are legal and make active response to hackers.

The second problem about remote loop is the transmission delay on Internet. Usually, the delay from source to destination changes from one millisecond to one minute. The remote sampling time should be larger than the variable delay. Communication intervals between higher level manager and lower level manager is about fifteen minutes in most NMSs, and this one is fit for the control model.

As for congestion or fault in channel, losing packet in Internet does frequently happen. In control systems, control commands and feedback information can't be lost, so how to ensure the control reliability is an important issue. SNMP bases on UDP which is an unconnected protocol and it's not reliable of course. Furthermore, NMSs in a large network are from multiple companies, how to integrate them with software is thus a challenge. Due to these considerations, SNMP isn't a good solution of remote control loop and CORBA is a better one. Since centralized network control is the basis of DNCM, it will be discussed in details later. Table 1 shows the terms mapping relationship between control theory and network management.

Table 1 Terms mapping relationship

Control theory	Network management
Sensor	Agent
Controlled object	Network devices
Reference value	Threshold
Feedback	Get/Trap
Control	Set

3. CENTRALIZED NETWORK CONTROL

Controlled objects in network control model are not only switches, but also some important servers, such as FTP, WWW servers. There are five vertical layers about network management in TMN. Some specifications about service layer management are put forward by Telecommunication Network Management Forum [3], which mean network service level management is required and necessary. Some research and experiments were carried out in IBM where admission control in Lotus file server with control theory was based on log files [4]. Compared with network element management level and network management level, service level management is much more closer to servers or workstations. It's well known that there're some differences in communication protocols supporting between servers and switches, therefore, the control system design will be discussed in two aspects.

Host service level control

During network management development, host management is always a hot area and a large number of RFCs are proposed by IETF [5]. In industry, Operation System (OS) companies built some private MIB for their own products, for example, WINS MIB by Microsoft. Some OSs provide powerful SNMP agents in OS. For example, there's an agent in windows 2000, which supports Host Resource MIB (RFC1514), LAN manager MIB II and Internet Service MIB. Some important

application servers have built special MIB, such as database MIB [6]. Communicating with agents in host, controller can get abundant feedback information which makes multiple objects control possible.

As for control commands transmission, hosts on Internet are all supporting TCP/IP. We are not only using SNMP Set primitive but also with other application layer protocols, such as CORBA or some self-defined protocols to complete control. All together, the principle of selecting control protocol is that the simple and general one is better. The host service control model is shown in Fig.6.

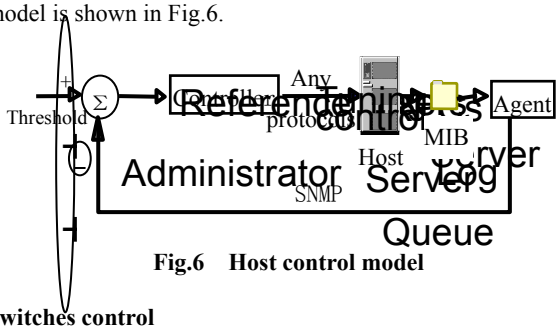


Fig.6 Host control model

Switches control

There're not screens, keyboards in switches or routers. Packets switching ability is the vital performance measurement of them. Two kinds of configuration ways are popular about switches. One is manual configuration with TELNET and the other is to set MIB variables with SNMP. TELNET has a long history and almost all OSs support it directly, so it's the main way of device control. Compared with configuration using TELENT, SNMP is unfit for network control for some reasons as below.

First, MIB information are only a subset of configuration fields with TELNET, furthermore, most of the variables are only readable. For example, there're more than one hundred commands in Cisco router and some commands can be used with different arguments. Then the commands and arguments combinations may be several hundred, covering configuration area from physical layer to application layer protocols. Second, for the network construction is a long process and not all devices support Set primitive in SNMP. It's hard to find a device which does not support TELNET. Third, SNMP isn't very secure because there is only one community to authenticate. With TELNET commands, there're usually multi authentications. The classical TELNET commands configuration process in Cisco router is listed in Fig.7 [7].

Step1: Login and do multiple times authentications.
Step2: Input commands.
Step3: Write memory. Write new configurations to memory and enable them be valuable right now.
Step4: Copy running-config to startup-config.
Copy the current configuration into file and they will be valuable for ever.

Fig.7 Cisco router configuration flow

Thereby, two approaches are possible in switch control system. First, adding new MIB; the alternative is to utilize TELNET commands instead of Set primitive in SNMP. Obviously the former is difficult to realize in engineering, because SNMP has been widely used and becomes the fact standard for computer network management. To protect the investments, network service providers disincline to change devices. In addition, SNMP is not appropriate when security is a significant requirement. TELNET commands approach is better in switches or routers control system. The detailed control model is shown in Fig. 8. Automatically controlling devices with software is introduced in another paper [8].

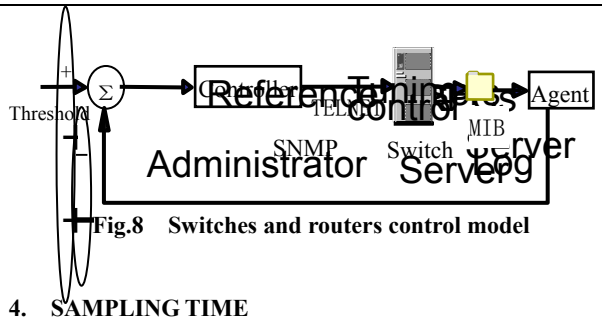


Fig.8 Switches and routers control model

4. SAMPLING TIME

Sampling time is the key factor for excellent control system. If it's too small, the bandwidth for control will be larger and the goodput of managed network will be reduced. If sampling time is too long, real time is not ensured. A fixed sampling time is easier for control system implementation than a variable one. Changing sampling time brings difficulties not only in computing but also in analyzing the stability of control system. Both Get primitive and traps are able to provide feedback information to network controller. Get primitive is executed every fixed interval by manager. But trap is an arbitrary event. Seven trap types are defined in RFC as Fig.9.

Two approaches can be used to do with trap. One is to receive traps but to wait for the next interval to respond. The other is to respond immediately. As for the former, sampling time is fixed and real time is discarded in control system. However, the later process real time traps but difficult to be realized in engineering. It seems that only through further study we can't draw any conclusion about which one is better. When devices start, restart or other serious faults happen, trap messages will be sent. In trap datagram, manager can't get any digital performance information. So trap has no effect on sampling time, but it can decide whether a controller should be shut down.

myLinkDown
coldStart
warmStart
linkDown
linkUp
authenticationFailure
egpNeighborLoss

Fig.9 Total trap events in SNMP

The special value of sampling time isn't easy to say in one word, because it depends on the agent self-sampling time and MIB variables. For example, agent samples IP packets flow information every one millisecond, but samples CPU loading information every one minute. The principle of selecting sampling time is that it must be close or a little more than the agent self-sampling time in local loop. Otherwise control system will generate redundant data which have no right physical meanings and waste network bandwidth.

The principle of the remote loop is discussed [10] and that it must be larger than the sum of agent self-sampling time and the Internet delay and retransmission delay.

5. ADMISSION CONTROL SYSTEM IN A GENETIC ALGORITHM SERVER

Admission control system

The hardware and software in the admission control system are distributed as shown in Fig.10. The controlled object is a genetic algorithm application server with double CPUs, running on Windows 2000 and using the SNMP agent provided by OS. The genetic algorithm server is used to

compute NP hard problems for users in our campus. Experiments show that every computing request occupies CPU resource about ten seconds. The detailed analysis about genetic algorithm has been shown in [11][12].

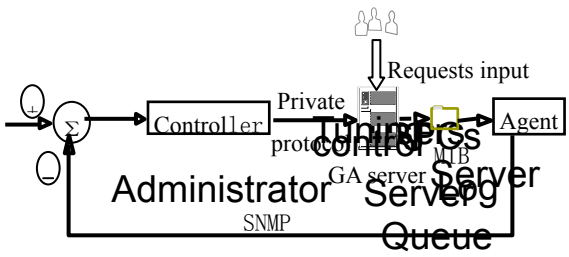


Fig.10 Admission control system

Genetic algorithm server tries its best to compute for users, but network manager hopes its CPU loading rate is between 60% and 70% for a long time to avoid high temperature and protect CPUs. In terms of the requirements, we select the MIB variable "hrProcessLoad" (the ASN.1 serial number is .1.3.6.1.2.1.25.3.13.1.2) to be the feedback information. Controller runs in the other PC in our lab and users send computing requests from Internet. Because all the devices and software process are distributed, it represents local host service control system.

Agent samples hrProccessorLoad every one minute and the sampling time of admission control system is also one minute. Controller decides the MaxRequests value and sends it to the application server with private control protocol. MaxRequests is the maximal parallel computing threads that can run in server.

Controllers

Three types of controller are designed and implemented in our experiments. Controller I is a very simple and control process is shown in Fig.11. MaxRequests(t) only depends on f(t) and has no relation to MaxRequests(t-1).

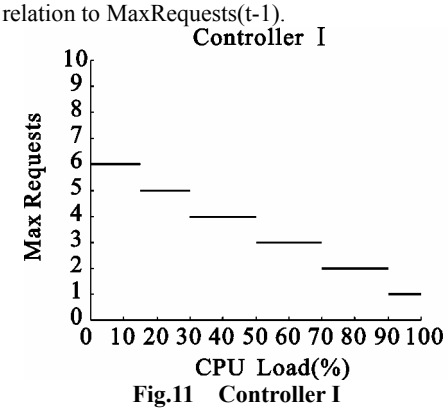


Fig.11 Controller I

In controller II, Maxrequests(t) depends on Maxrequets(t-1) with multiple function. The controller equations are as below. f(t) is the feedback data at t time, and α (t) is the coefficent function. When application server works on the expected status, α (t) will be one which leads the MaxRequests(t) to be equal to MaxRequest(t-1), otherwise α (t) will change. [] is the round operation. It's possible for MasRequests to be zero after round operation, and zero multiplied by any number is still zero. So we define that MaxRequest will be one if the round operation result is zero.

$$Max\,\mathrm{Re}quests(t) = [\alpha(t) \times Max\,\mathrm{Re}quests(t-1)]$$

$$\alpha(t) = 1 - (f(t) - 0.60) = 1.6 - f(t)$$

In controller III, MaxRequests(t) and MaxRequests(t-1) is an addition relationship. The detailed equations are as below. Based on the control theory, system stability is closely related to the argument k. If k is bigger, real time feature is better but jitter is obvious. To get a good k and MaxRequests(0), three combinations are tried and three kinds of distinguished control effects are got.

$$Max\,\mathrm{Re}quests(t) = [Max\,\mathrm{Re}quests(t-1) + k \times e(t)]$$

$$e(t) = 0.6 - f(t)$$

Control effects

The input request rate is as square wave in Fig.12.

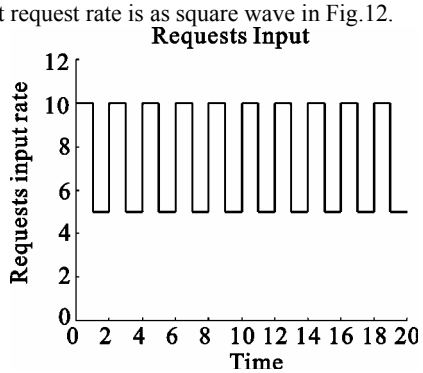


Fig.12 Input requests

The period is two minutes, in the first half sampling period, there're ten requests per minute and in the second it's five. The average input rate is 7.5 requests per minute and exceeds the service computing ability. Fig.13 is an unexpected server occasions which there's not an admission control system.

With controller I, application server works as Fig.14. Although controller I server works better, it's also unacceptable. For most of the time, the CPU occupancy is between 70% and 100%, overshooting the expected 60%.

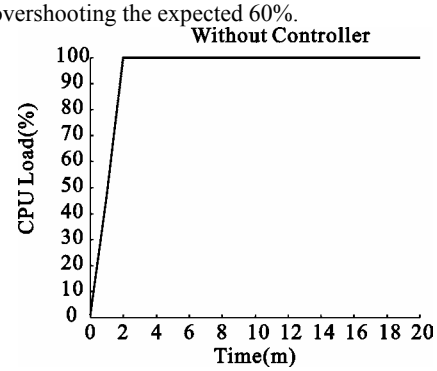


Fig.13 Server status without controller

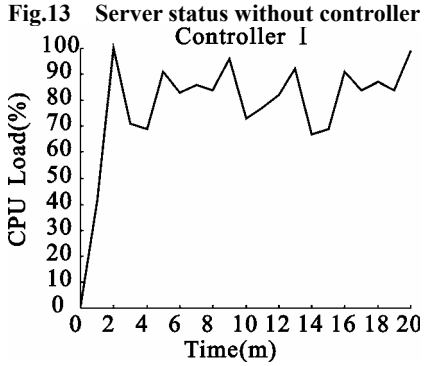


Fig.14 The effect of controller I

Fig.15 shows the control effect of controller II. After six minutes, it becomes stable and works in the expected status. Controller II is a good controller for the admission control system, but six minutes may be long for the control system when the MaxRequests(0) is set six. If MaxRequests(0) is set smaller., the setting time will be reduced. From it, we can see that the control system will achieve good control effect in engineer. Especially the model is easily to be implemented in practice due to simplicity of mathematical formulation.

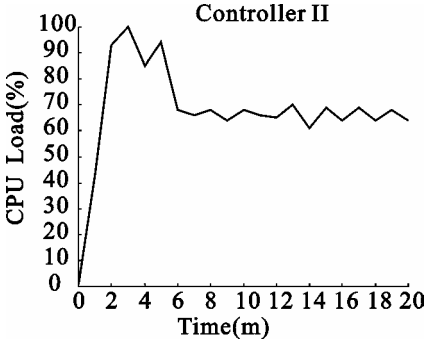


Fig.15 The effect of controller II

When MaxRequests(0) is six and k is six, the control effect of controller III is shown in Fig.16. The experiment result is shown in Fig.17 when MaxRequests(0) is zero and k is six. Clearly the jitter in Fig.16 and Fig.17 is not acceptable. Let's reduce k and set it three, then we get the control effect as Fig.18. Now this one is also a good one.

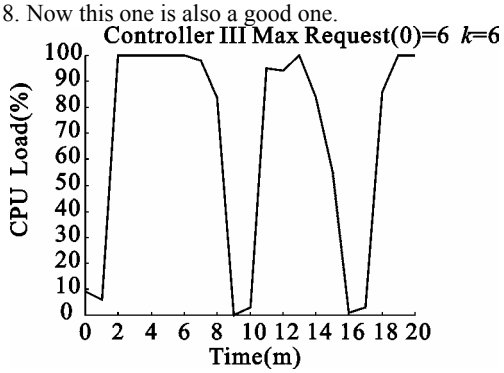


Fig.16 Control effect of controller III

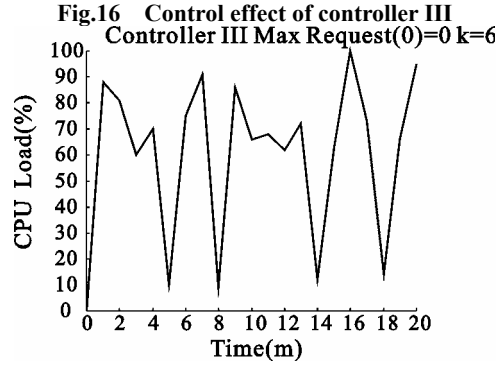


Fig.17 Control effect of controller III

From the design of controller III, we can see control system is not only related to controller equations but also closely related to the arguments value. Selecting good arguments value is a difficult task in both theory research and control engineering. So choosing simple equations with a few arguments or classical control equations may be a good way for network control system design.

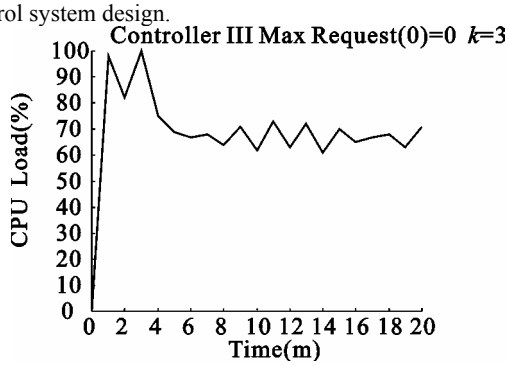


Fig.18 Control effect of controller III

6. CONCLUSIONS

On the basis of network management model, we have introduced a procedure of designing the network controller. Under network control model, NMS can realize network monitor and network real time control according to the monitoring information. This is the requirements of current NMS and future work of NMS research. With the SNMP and control theory, we can accomplish complicated and digital network performance management, and even more complicated network management will be carried out, such as network performance data mining, network action prediction and control and so on. With the novel approach in this paper, NMS will be more powerful and provide the QoS of managed network to some extent.

Since automated network management with SNMP and Control theory is put forward first in this paper, the control experiments are not very difficult and the classical control methods are not used, such as Z transform, Laplace transform and stability analysis with mathematical tools. It's a research area deserves more attention in the future.

7. REFERENCES

[1] Srinivasan Keshav. A control-theoretic approach to flow control. In Proceedings of ACM SIGCOMM '91, September 1991.pp.3-16

[2] Lotfi Benmohamed and Semyon M. Meerkov. Feedback control of congestion in packet switching networks: the case of a single congested node. IEEE Transactions on Networking, 1(6), December 1993. pp.693-708

[3] Telecommunication Management FORUM TMF 508v3.0 Connection and Service Management Business Agreement, April 2001

[4] Sujay Parekh, Neha Gandhi, Joe Hellerstein, Dawn Tilbury, T. S. Jayram and Joe Bigus. An Introduction to Control Theory and Its Application to Computer Science. SIGMETRICS 2001/Performance 2001

[5] P. Grillo, S. Waldbusser. RFC 1514: Host Resources MIB. September 1993

[6] K.McCloghrie，J.Galvin. RFC1447: Party MIB for version 2 of the Simple Network Management Protocol (SNMPv2). April 1993

[7] Cisco Content Router User Guide. Available at www.cisco.com

[8] Yu Yijiao. Automated Network Management [Master thesis] Central China Normal University, Wuhan 2002

[9] M. Rose. RFC1215:A Convention for Defining Traps for use with the SNMP. March 1991

[10] Yang,S. and Alty,J.L. Development of a Distributed Simulator for Control Experiments through the Internet. Future Generation Computer Systems. 18(5), 2002. pp595-611

[11] Yu yijiao, Hopfield Neural Network and Genetic Algorithm in Solving Traveling Salesman Problem: Experimental Comparison and Analysis, Journal of Central China Normal University (Natural science), No.2, pp157-161, 2001

[12] Yu yijiao, The Analysis about parameters in solving TSP with genetic algorithms, Journal of Central China Normal University (Natural science), No.1, pp25-29, 2002

Agent in Electronic Commerce

Qianping Wang, Ke Wang, YiCai Xie
School of Computer Science and Technology, China University of Mining and Technology Name
XuZhou, 221008, China
E-mail: qpwang@cumt.edu.cn

ABSTRACT

E-commerce can be described as the process of conducting trading or facilitating traditional commerce activities via the use of the Internet. Using the Internet as a platform to deliver novel services and for conducting business processes has become one of the most discussed topics in the popular press. In this paper, it is analyze how agents may facilitate a variety of relevant e-commerce functions, such as advertising, matchmaking and brokering. It is presented that an approach to build institutions of agent-mediated trading within the Internet.

Keywords: Electronic Commerce, Agent, Intelligence Agent, Multi-agent, virtual enterprises

1. INTRODUCTION

The electronic marketplace provided by the Internet and Web is about to establish itself as a significant economic factor worldwide. It does not only leverage relationships for strategic advantage in global commerce, it also enables the formation of virtual enterprises. These virtual enterprises are capable of producing and delivering value-added services and products tailored to individual just-in-time customer preferences [1]. On the other hand, it raises customer expectations regarding issues of new quality of shopping experience and product and service quality of new brands on the Internet. This demands for advanced solutions, not only for the rather traditional business-to-business/customer electronic commerce, but also for integrated commerce in the future.

Intelligent agents are autonomous computational software entities which have access to multiple, heterogeneous data and information sources, and pro-actively acquire, mediate, and maintain relevant information on behalf of their users or other agents. The agents are especially meant to offer value-added information services and products. That includes retrieving, purchasing, filtering, fusing, and presenting relevant information to human decision-makers on demand and preferably just in time. Though e-commerce on the Web is not the classical application domain of agents, it certainly is the most steadily growing one. E-commerce application is expected to be more convincing, more attractive, and more qualitative for the common users' business on the Web.

2. E-COMMERCE

The start of the electronic information service was dominated by a small number of powerful service providers and service delivery channels. These offered a relatively fixed product and service range. A poor match between the user requirements, perceived by the suppliers before production, and the actual user requirements impaired service uptake. Customers often needed to modify their business models and requirements substantially to fit them to the information and communication products on offer. Under pressure from different market forces such as increasing privatization, diversification and competition, the customer beholds increasing choice, complexity and heterogeneity at a variety of levels including types of message transport, portal, service and customer interfaces.

Consumer-driven customizable service

Customization can be defined as the ability of a product or service to be modified and maintained to meet particular customer requirements or profiles. Customization can occur at multiple levels and can be customer, third-party (e.g. broker) or provider driven. Here customer (or consumer), broker, and provider, are just temporal roles played depending on an entity's position in the supply chain for a particular service [1]. A new service can be synthesized from combinations of services from different vendors, resulting in a cost reduction. This seeds a growth in third-party intervention offering independent advice to users to help them intelligently search and select service combinations. A broker may be able to simplify, configure and coalesce access to multiple services for the user. For optimal customization, service providers need to produce interfaces and delivery channels with suitable designs and abstractions to support reconfiguration and cooperation.

User requirements for a highly customizable service

We now discuss several specific user requirements that support a highly customizable service, extending the requirements of distributed systems such as openness and scalability. These comprise:

- Accessible, configurable and secure service profiles
- Accessible, configurable and secure customer profiles (also called personal profiles)
- Sophisticated advertising and matching of service provision to service requests
- Cooperation, coordination, control and coherence of groups of autonomous service providers and users.

Service profiles are a high-level publicly configurable form of the service interface that is distinct from the internal developer interfaces. Service profiles define which elements of the server can be modified. Although service interfaces may be available, they may not be habitable; for example, they may not be configurable by the user, users may not understand how to use them, or they may be too low-level. Habitable services provide interfaces at the correct level of abstraction and modularization.

The companies need to shift from the old world of mass production where "standardized products, homogeneous markets, and long product life and development cycles were the rule" to the new world where "variety and customization supplant standardized products". So companies need to be able to, at a minimum, develop multiple products that meet the multiple needs of multiple customers.

Electronic commerce is defined as the set of activities of trading goods and services online. It can be structured into the market segments: business-to-customer (B2C), business-to-business (B2B), customer-to-business (C2B) and customer-to-customer (C2C) e-commerce.

The movement toward E-commerce has allowed companies to provide customers with more options. However, in expanding to this new level of customization, businesses increase the amount of information that customers must process before they are able to select which items meet their needs. One solution to this information overload problem is the use of agent system that recommends something you need.
According to recent market research reports, up to 71% of companies worldwide are expected to link to the electronic marketplace on the Internet, generating up to 20% e-business-based turnover on average by 2004 [1]. Basic key technologies for the development of e-business solution include:standard data representation, retrieval and exchange

- secure user profiling and data,
- secure electronic payment,
- standard protocols covering most issues of electronic trading,

In order to apprehend e-commerce in a more homogeneous way a variety of different architectures, frameworks, and reference models have been developed and are under ongoing research. These include, for example, the CBB (consumer-buying-behavior) model developed at MIT. However, each of these approaches appears to be either too generally specified or too specific in part; an efficient, dynamic refinement of interfaces and service components by the user is, if at all, scarcely supported.

3. INTELLIGENT AGENT

An agent is a piece of software capable of acting intelligently on behalf of a user or users, in order to accomplish a task. Agents, like humans, co-operate so that a society of agents can combine their abilities to resolve problems. Agent technology is currently being applied to a number of application areas including network management, information filtering, user modeling, and business process management.
A customer requests their agent to find a particular type of goods, within a specific price range, with a choice of supermarkets, for a particular set of species. The agent searches for all available agents offering retail services, requests each to provide information on the goods that match its criteria, ranks the suitable goods based on its knowledge of the user's preferences, and finally reports back to its user. The user selects their preferred goods and authorizes the agent to agree a contract and pay the required deposit.
The sample contains two types of agent — customer agents which represent the interests of a customer, and seller agents offering retail services and representing the interests of retailers.
Each agent would be owned by and represent a different party. Similarly, each agent could have been developed by different companies, at different times and using different technologies. The agent provides:

- a commonly agreed means by which agents can communicate with each other so that they can exchange information, negotiate for services, or delegate tasks,
- facilities whereby agents can locate each other,
- an environment which is secure and trusted where agents can operate and exchange confidential messages,
- a means of interacting with users,

Software agents
The software agents are just independently executing programs which are capable of acting autonomously in the presence of expected and unexpected events. Agents can be of differing abilities, but typically possess the required expertise to fulfill their design objectives. To be described as 'intelligent', software agents should also possess the ability of acting autonomously, that is, without direct human input at run-time, and flexibly, that is, being able to balance their reactive behavior, in response to changes in their environment, with their pro-active or goal-directed behavior; in the context of systems of multiple autonomously acting software agents, they additionally require the ability to communicate with other agents, that is, to be social.
The central concept that distinguishes software agents from simple programs is their interaction with their environment; typically, we say that an agent is embedded or situated within its environment. While it is true that any distributed system could be implemented as a single centralized system, such an approach ignores the fact that, in some types of environment, information and control are naturally distributed throughout the system; this is particularly true of communication systems. Distributing agents to deal with information and control locally, while enabling communication at higher levels of abstraction, can make systems easier to understand and hence make them easier to design and develop. In this sense, agents can be embedded throughout a system performing functions with appropriate levels of expertise.
Agents can act as intelligent decision-makers, active resource monitors, wrappers to encapsulate legacy software, mediators, or as simple control functions. The common link between this vast array of possibilities is that each agent autonomously strives to meet its own design objectives. In some ways, agent-based computing can be seen as a natural extension of object-oriented programming whereby objects are empowered with their own thread of control and their own goals or objectives, so that they autonomously control their own behavior in order to reach their goals.

Intelligent Agent
We define an agent as an autonomous, computational software entity that has access to one or more heterogeneous and geographically distributed information sources, and which pro-actively acquires, mediates, and maintains relevant information on behalf of users or other agents preferably just-in-time. Thus, an agent is supposed to satisfy one or more of the following requirements:

- Information acquisition and management.
- Information synthesis and presentation.
- Intelligent user assistance.

Many agents have been developed or are currently under development in academic and commercial research labs, but they still have to wait to make it out to the real world of Internet users broadly.
According to the definition of agents we can differentiate between communication, knowledge, collaboration, and rather low-level task skills.
Today, intelligent agents are deployed in different settings, such as industrial control, Internet searching, personal assistance, network management, games, software distribution, and many others.
Agent technology is quite on its way to producing mature standards concerning software agent architectures and applications, such as OMG MASIF (mobile agent system interoperability facility) and FIPA's agent-related specifications.

The FIPA agent standards
Agent standards have reflected the distinction between stationary and mobile agents: the Foundation for Intelligent Physical Agents (FIPA) and the Object Management Group's Mobile Agent System Interoperability Facility (MASIF).

OMG's MASIF differs from FIPA in that it regards the mobility of an agent from one location to another to be its defining characteristic. MASIF defines interaction in terms of Remote Procedure Calls (RPC) or Remote Method Invocation (RMI); it does not support or standardize interoperability between non-mobile agents on different agent platforms. Further, MASIF restricts the interoperability of agents to agents developed on a single type of platform whereas the focus of FIPA is to directly support the interoperability of agents deployed on heterogeneous agent frameworks. Stationary and mobile agent standards are, however, starting to overlap and may ultimately converge [4].

There are several versions of the specifications for FIPA standards in existence. Currently the most widely implemented version is FIPA 97 Version 2.0. In practice the two core parts are: Part 1, which defines the agent platform or agent reference model, and Part 2, which specifies the agent communication language. The FIPA 97 agent reference model (Figure 1) provides the normative framework within which FIPA agents exist and operate. Combined with the agent life cycle, it establishes the logical and temporal contexts for the creation, operation and retirement of agents [1].

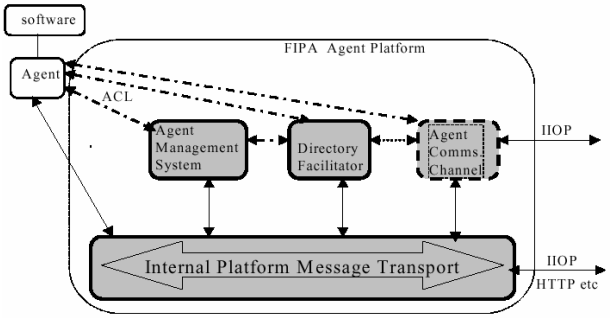


Fig.1 The FIPA 97 agent reference model

The FIPA standards in some areas introduce conceptual problems for designers and implementers. For example, the FIPA Agent Communication Language (FIPA ACL) focuses on an internal mental agency of beliefs, desires and intentions, and closure is not enforced (agents are not compelled to answer); both these concepts hinder multi-agent coordination.

The FIPA ACL describes a standard way to package messages, in such a way that it is clear to other compliant agents what the purpose of the communication is. Although there are several hundred verbs in English corresponding to performatives, the FIPA ACL defines what is considered to be a minimal set for agent communication (it consists of approximately 20 performatives). This method provides for a flexible approach for communication between software entities exhibiting benefits including:

- Dynamic introduction and removal of services
- Customized services can be introduced without a requirement to recompile the code of the clients at run-time
- Allowance for more decentralized peer-to-peer realization of software
- A universal message-based language approach providing a consistent speech-act-based interface throughout software (flat hierarchy of interfaces)
- Asynchronous message-based interaction between entities.

Multi-agent systems

The ability of an agent to be social and to interact with other agents means that many systems can be viewed as multi-agent systems (MAS). Our definition of software agents as autonomous processes means that they encapsulate their own state and behavior. For agents to interact, they must also possess the ability to communicate with other agents via some common agent communication language (ACL). Similarly, for agents to cooperate with one another or to coordinate activities, requires some inherent desire to be social, whether this is due to self-interested, utility-maximizing criteria among competitive agents, or because the system as a whole is designed to achieve some ultimate goal, through essentially cooperative agents pursuing the common good. Any such MAS must include infrastructure to support transport and directory services among agents, ACL message encoders and parsers, dialogue management, and brokers and facilitators.

Multi-agent systems can adopt a range of sophisticated approaches for advertising services, customer preferences and subsequently matching service usage to service provision through interaction between service, user and facilitator agents. Agents can deploy a variety of strategies for the cooperation, coordination, control and coherence of groups of services and users. For example, service and user agents can send messages within the context of interaction sequences such as client server interactions, auction mechanisms, the contract net protocol, and different types of brokerage to support complex negotiation strategies. Multiple providers and users who share common service and customer ontology have a basis to gain a much deeper agreement in the provision and use of services.

4. AGENT IN E-COMMERCE

It has been argued that in the perfect electronic market, buyers and sellers will be able to contact each other in a direct manner, thereby "eliminating the middleman". However, evidence in the marketplace demonstrates that at least for some time to come, the role of agent is becoming increasingly important. The role of agent has been discussed, and the use of agents for this task is surveyed.

In an open and increasingly commercialized cyberspace, personalized agents not only may pro-actively discover and manage information relevant to its customers but also be paid and have to pay for any services they provide. Reasonably, the agents have to be equipped with effective and efficient methods for making rational decisions. The agents may make purchases up to a pre-authorized limit, filter information and solicitation from vendors, trade products and commodities such as even bandwidth and components within the B2B (B2C) digital market, decide on bids from service providers for its customer in some reverse auction online in a consumer-to-business e-commerce setting, and increase the level of trust in their actions gradually over time, involving only manageable risks for both customers and vendors.

Basic for agent-mediated E-commerce

In addition to the techniques for developing e-commerce solutions for many negotiation schemes and trading mechanisms in agent-based systems rely on:

. multi-attribute utility, price comparison, content-based recommendation and so on,

. collaborative recommendation, coalition formation among agents, service matchmakings and brokering among agents, and arbitration schemes in case of agents with conflicting interests.

While the first of these mechanisms are mainly used in the domain of shop-bots, the later are being used for multi-agent systems where different agents are provided with a common interaction standard and strategies. The choice of standard depends on what properties the system designer wants the multi-agent system. The standard for agents can be evaluated according to different criteria such as: the sum of all agents'

payoffs in a given solution, an agent's payoff is at least as high as it would get, stability of the mechanism, as well as communication and computational efficiency.
Agents may perform reasoning using quantitative utility functions in ways well known from decision-making, especially in computing optimal and assessing their sensitivity. A variety of methods have been invented to allow agents to determine which negotiation strategy is more successful, for example, using evolutionary computing, fuzzy rules and so on.

Agents on markets
Agents may collaborate for gaining or sharing benefits, though the degree of rationality or collaboration depends on the outcome of the negotiation. Such negotiation among agents may impact, the charges for services provided as well as the kinds of services or goods themselves. Free markets are the commonest virtual institutions for e-commerce to be mediated by collaborating agents; they are the means for C2C and B2C e-commerce, respectively. Markets provide locations where multiple agents of customers and vendors may meet to negotiate and exchange information. A Good example for agent-based marketplaces is virtual marketplace. It relies on a multi-agent infrastructure with agents that buy and sell services from each other using a given set of commerce.

Seller agents
Seller agents are used by E-commerce to suggest products to their customers. The products can be recommended based on the top overall sellers on a site, based on the demographics of the customer, or based on an analysis of the past buying behavior of the customer as a prediction for future buying behavior. Seller agents automate personalization on the Web, enabling individual personalization for each customer. Many of the largest commerce websites are using seller agents to help their customers find products to purchase. A seller agent learns from a customer and recommends products that she will find most valuable from among the available products. The seller agents that present to customers, the technologies used to create the recommendations, and the inputs they need from customers.

Agent for searching and browsers of buyers
One major function of a seller agent is to facilitate searching in a market-space. Some market-spaces are very large and help in sorting through the options is required; others offer very complex product offerings and help in matching product offerings and business needs is useful. The search function of the seller agents has been widely recognized, and has been called either making searching easier or reducing search costs. Otherwise, visitors to a web site often look over the site without ever purchasing anything. Seller agents can help customers find products they wish to purchase.

Fig.2 Browse products

In traditional commerce a customer might walk into a store and ask the clerk to recommend goods. Ideally, the clerk would recommend several goods, and the customer could go to locate the recommended goods, browse the appearance, and see which ones appealed to them. The quality of the recommendations provided was dependent on the particular clerk's knowledge of an enormous range of goods. Some E-commerce site has several advantages when implementing browse by seller agents. First, the seller agents combine knowledge of from several clerks so that higher quality recommendations can be provided no matter what the query parameters. Furthermore, seller agent return with immediate links to the items being recommended – no more searching the store for the obscure recommended. Browse of seller agents helps the E-commerce site by converting browsers into buyers.

Agent for content-based recommendation
Recommendation agent focuses on identification of similar users and their opinions to recommend items [2]. It is a powerful method for leveraging the information contained in user profiles. In contrast to content-based filtering through which items are recommended to a user according to correlations found between the items' descriptions and the given or observed users preferences stored in a profile, an agent rates items chosen by its user and compares the corresponding user preference vector to that of other known users. It then recommends other items that have been recommended by users who share similar likes *and* dislikes. For this purpose it has to collaborate with other agents to gain the respective knowledge.

Fig.3 recommendation from CISCO

As a seller of attribute-based
The seller agent sends information about the product, rather than purely price information. In particular, analysis should be done on the cost of product information relative to the cost of the price information. Attribute based seller agents will recommend products to customers based on syntactic properties of the products [3]. For instance, if the customer does a search for a romance hat, and the seller agents responds with a list of several recommended hats. Attribute-based seller agents are often manual, since the customer must directly request the recommendation by entering his desired product properties. Attribute-based seller agents can be personal, depending on whether the E-commerce website remembers the attribute preferences for customers. The seller agents are entirely based on the category of goods the customer selects. Since customers must explicitly navigate to a category to obtain a suggestion.

Fig.4 Attribute based recommendation

Customer Online Comments

The customer comments feature allows customers to receive text recommendations based on the opinions of other customers. Located on the information page for each book is a list of 1-5 star ratings and written comments provided by customers who have read the book in question and submitted a review. Customers have the option of incorporating these recommendations into their purchase decision.

Fig.5 Online Comments

Dynamic nature of seller agent

Another aspect of the seller agents is the dynamic nature. That is agents operate in a dynamic environment. This dynamic nature of the seller agents makes it very difficult to solve analytical. Many small, simple agents working together in even a simple environment can lead to complex system behavior. It is included the dynamic system behavior as well, such as how many buyers and sellers will opt to use the agents, what supply and demand will be observed in the market-space.

5. CONCLUSION

Using autonomous trading agents may have different impacts on Internet-based economies and business in the future. Such agents may make purchases up to an authorized limit, filter information and solicitation from different merchants, and dynamically trade any type of good pro-actively on behalf of its users in markets. Intelligent agents are also driving research and development of related technologies such as generation and reuse of ontology for electronic commerce. The agents are creating value for E-commerce websites and their customers. And agents can then afford to charge relatively high fees to buyers or sellers and still maintain a large volume of transactions and gather high revenues. The agent needed to support sellers, buyers for different strategies implied by the system. The real agents can perform searching, delivery or payment, and many other functions.

Other future research of agent-mediated electronic commerce that encompasses agent-based coordination of the whole supply chain associated with any product ordering by a customer, and collaborative customer relationship management.

6. REFERENCES

[1] Alex L.G. Hayzelden, Rachel A. Bourne Eds, “Agent Technology for Communication Infrastructures” John Wiley & Sons Ltd, 2001, New York

[2] C J. Ben Schafer, et al, “Recommender Systems in E-Commerce“, Proc. Of E-COMMERCE 99, pp158-166, Denver, Colorado.

[3] Arie Segev et al., ”Brokering Strategies in Electronic Commerce Markets”, Proc. Of E-COMMERCE 99, pp167-176, Denver, Colorado.

[4] P. D. O’Brien and R C Nicol, ” FIPA — towards a standard for software agents”, BT Technol J Vol 16 No 3 July 1998.

VOD--most potential application on broad band network

Chen wei
The department of Information,
Wuhan university of technology
Wuhan Hubei , china
Qiyan
The department of Information ,
Wuhan university of technology
Wuhan Hubei ,china
E-mail:San1113@sina.com

ABSTRACT

VOD will be mainstream of accessing message in the future. it has potency to develop. In this paper, we will introduce the two main components of VOD: video server and client terminal unit and their corresponding technology and its' tendency .

Keywords: distributed parellel video server, VOD, set-on-top box, stream ,coding

1. INTRODUCTION

VOD, that is video-on-demand, wherever you are, turning on the TV or computer or using your mobile terminal, you can see your favorite program at any time .what the information technology want to do is meeting the demand through multimedia broad band network. Comparing with other applications on broad band network, VOD is the closest service to people's life. On the other hand , it is difficult to realize. In china telecom's opinion, we have already had high-speed path---broad band network, VOD then will be the most potential application.. VOD will be mainstream of accessing message in the future. it has great prospect.
The system of VOD is made up by four parts: video-server, communication-network, client terminal unit and management system.

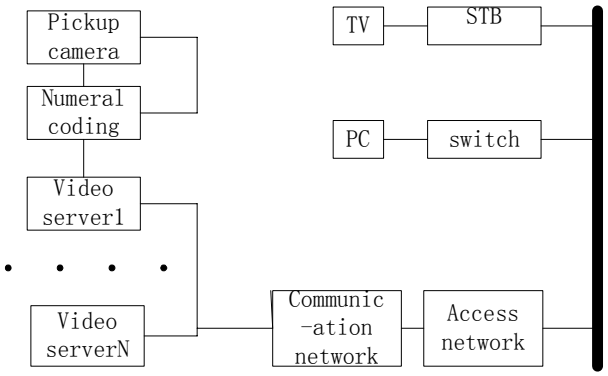


Figure 1 The system of VOD

2. VIDEO SERVER

One common architecture shared by traditional VOD system is single-server model , however, the single-server has limitations of expansibility and fault tolerance. So distributed parellel video server has more prospects. It can more easily realize the balance of load.

2.1 The structure of Parellel video server

The base of DPVS is video-data -striping server .the anticipated data gets its' own video stream to play, so every video data striping from server must be recognized and fabricate through proxy. Proxy can be realized through software or hardware. This demands identifying the quantity and address of server, as well as the location of video data and tactics of band division. There are 3 types of proxy:
Proxy-at-server, independent-proxy, proxy-at- client. moreover, proxy can use redundant data to mask the server error, to extend the server fault tolerance.

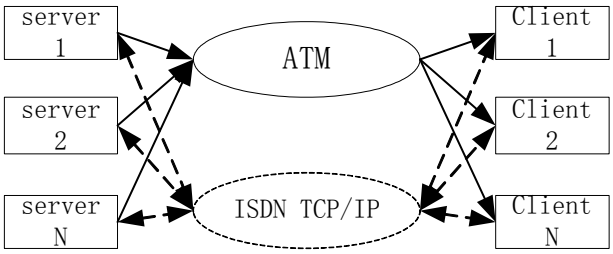


Figure 2 Network of VOD

The distributed parallel server makes use of low-port computer as segment of server
Network to improve server level. It can offer server on all kind of network. The system of two-layer video server can be used on the network of corporation and hotel. Video server which has more layers can be designed as a large scale video server. We can install a video file server in every tenement, install one-layer file server on district formed by some tenements, install two-layer or multi-layer file server on several districts. All those come into being a huge network system which can storage large quantity of data.

2.2 There are 3 types of structure of video server

Current mainframe : The earliest video server adopts mainframe which runs on the standard operation system. Such as unix system. The hardware is made up by a lot of array of video disk. The main function of video server is storaging, choosing and conveying data, seldom dealing with data.
Multi-function processor: Video server is fabricating a large number of hardware units which can accomplish some directive, comb-
-ining those relevant units as corresponding system. Some systems good at creating immo-
-bility image, or managing database, or network equipments, or video database. All those systems combines as multi-function video server .
Tuning-in video server: Tuning-in video server is made by a special emulator. Disk controller, ATM and counter computer interface can also use the emulator to communicate. The server can extend the capacity easily.

2.3 The key technology of video server

Parellel managing the distributed stream of video: distributed model, supporting the group of video servers, large quantity of video data streams parellel. Managing video database: the space of video disk base of service system is limited, for example ,if the storage space of program is 800M/h,diskbase of 150GB can only storage about 100 programs. The video database management system can delete or edit some programs.

The vase neck of video server: VOD is very different , it does not require high-speed CPU, but is very strict with the disk array. Take the fastest disk ---160M/S ULTRA 3 SCSI for example, through mainframe I/O, the speed is at 60% off, how many parellel stream of 1.5G can IT support ?

THE balance of load: In large scale district, there may be hundreds of parellel stream, at that time, we should need two more servers working together distributed managing the stream. Commonly, we have two solution, one is dividing the network by band, it means that dividing the users into different segment by band, such as the segment of 192.168.x.x and the segment of 10.10.x.x . Different users connect with their own server. The other solution is balancing the group servers's load, it means that establishing a group servers system which is in charge of the equipments balancing the load and allocating the load dynamic.

Using video buffer pool to improve the VOD system: If thousands of users access your video at the same time, I think your conveying speed may be at hundred GBITS/S. How to solve it? Founding a video buffer pool around the video server. now you can provide video service to thousands of users only using a small server simultaneously. On the distributed broad band network model, many video servers as small scale server distribute in the base installation of network transmission .

For example, on the fore port of cable or the hub of network based on HFC, every video server can only control one subset of program memory, support one subset of video stream. The basic installation of this network require enough band which can connect the server's hub with every segment.

Technology of operating stream: this technology can support conveying hundreds of high-quality multimedia stream into the client computers. The client terminal can play any multimedia program in the server's storage at any time. It also allows that when the client receives the useful data, he can stop waiting and watching the program immediately. This technology hasn't the defect of long-time waiting or out of controlling .It can adjust the work of the system in order to adapt to the communication of the network, for multimedia server, offering high quality service and short-time startup is more important. In fact, the technology of operating flow is the key to meet this request.

Striping policies: When single video disk cannot accommodate enough band for a video program, we should add the band width of the storage system in order to support the video data stream. one solution is making use of RAIDs as the basement of memory equipment. In accordance with the accessing model , the data needed by single request is provided by the RAIDs disk. Another choice is that using the striping based on the software. The striping policies combine many disk drivers as one logic disk driver. In another words, it distributes data liking stripe on many disks. At any time, we can request data storaging on one logic disk, as well as conveying at summation of all disk band. The policies of striping can identify the configuration of strip, directly access disk equipment. so the software of server can automatically balance the video flow in terms of its' load, ensure every client receive high-quality program .

Code processor: MPEG-4 technology will apply into many equipments .Now transplanting the MPEG-4code processor into DSP ,which can process data 10-50 times faster than the fastest processor ,achieve MPEG-4 code processor. Using MPEG-4 code processor, we can get high-quality and low-storage-space program.

Real time medium code processor with flow model on internet: connecting it with VTR or other A/V signal ,we can switch into MPEG-4 format. Receive all types of video and frequency data, also input stream through internet interface. Moreover, we can convey many types of data simultaneously. supporting on-line broadcasting on internet.

2.3 Client terminal equipment

Digital web set-top-box: The set-on-top adopts 32bit RISC CPU and the operation system platform, equipped with advanced photo processor and reliable system software and abundant application software, supports WAP protocol. We can enjoy the high-quality video program.Set-top-box is a very important facility .Till to now , the function of it has extended from a multi-frequency tuned-in to database controller terminal. The open model ITV set-top-box is made up by microprocessor, digital concoct, ADSL interface, NTSC/PAL UNIT , ROM/RAM and extended interface.

3. TECHNOLOGY OF VOD ACCESS NETWORK：

the access network of client terminal classify seven ways: MODEM,N-ISDN,B-ISDN,ADSL,HFC,FIBER,SATELLITE. In the following, we will give a chart about the VOD access network.

4. THE KEY TECHNOLOGY OF VOD SYSTEM

The concept of stream medium: Streaming is also called flow model medium. It means that delivery the program into the internet as data package through video conveying server, The client terminal recover those data.The package during the courses is named stream. The key technology is streaming. Referring to the technology of carrying data through internet.

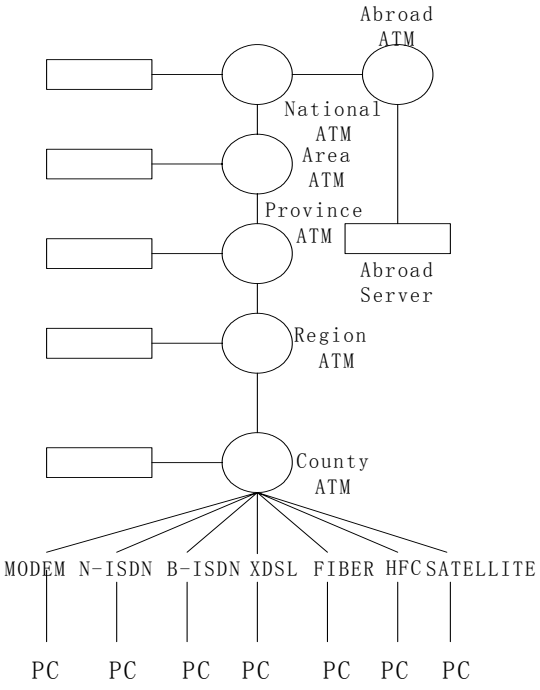


Figure3 VOD access network

There are two methods of realizing streaming:Realtime streaming and progressive streaming. On the average, if the video is realtime broadcasting , using streaming server or RTSP realtime protocol, that is realtime streaming. If using HTTP server, data is progressive streaming .of course, the streaming file also supports loading on the disk before playing .

The method of stream medium transmit: there are three transmit technology :

unicast, multicast, broadcast.

Multicast is called groupcasting. The character of the unicast is that the resource and the destination of the stream is corresponded one by one. That is, stream sends out from resource can only arrive at one destination. The groupcast is broadcasting based on groups, one resource (the server) correspond several destination(the client),but all those should be strict in the group. The broadcast , his resource arrives at corresponding several destination,

but the relationship cannot be strict in the same group, that is ,when the stream sends out ,all destination(client)can receive at the same band segment. so the broadcast can be seen as a special example of multicast. Digital video and the transmission technology of voice. At present, among the large number of video codify algorisms, the MPEG and H.26X have great influence and apply widely. The basic theory of video compression is the theory of information. Compression means that getting rid of the redundant information from the area of time and space.

MPEG-1 aims for compressing the image of SIF standard rate of differentiation (for NTSC is 352*240, for PALis 352*288),at the transmission rate of 1.5Mbits/s,max to 4.5Mbits/s, has the quality of CD, equal to VHS.

MPEG-2 aims for higher quality picture of advanced industry standard and higher ratio of transmission. it can provides transmission ratio about 3-15Mbits/s, the ratio of differenti--ation is 720*486 on NTSC.

MPEG-4 improves the algorism of com--pression.. Adopt the conception of matter, the ratio of compression is 100 times than it before the high-level voice and the video.MPEG-4 can meet the need of multimedia communication at low ratio.

MPEG-7 is also named multimedia content description interface. It want to describe the relationship between structure and the standard. it pays more attention to integrate the computer vision and database and DSP. It can be used both in memorying and the streaming.

MPEG-21, now we are going to study it from six aspects: the client demand, interactive program, program expression, program recognition and description, IPMP technology and terminal technology.

H.263code: H.263 code is a mixed method which can reduce redundancy of frame prediction .It can offer better image at lower speed, suitable for IP video meeting , Be absorbed by many communication terminals, such as :

ITU-TH.310(B-ISDN)、H.320(ISDN), H.324(PSTN)H.323(LANORWAN,INTERNET)

5. APPLICATION

VOD serves telecommunication, through backbone network, picking saloons and district up, offer VOD /SOD service, NVOD service/digital television service /digital saloons/educationVOD/multimedia-lesson-on-demand/library VODsystem/bookstoreVOD/MOD system.

6. CONCLUSION

This is revolution of video information, as well as the revolution of broad band network. we can realize the network which can transmit video information as well as the voice and data. The focus on combining the three network into one means that establishing broad band which can carry voice data and video. The most difficult trouble in realizing VOD for VOD is that it needs more band width, in another word, VOD has more potent to improve. Digital information era is the future's tendency.

7 . REFERENCES

[1] Xu jijin, Tan hang.“Discuss about The application of VOD”,Journal of Telecom Science ,No 7 ,1996

[2] Shen zhengyuan. “The foreground of VOD ”,Journal of Micro-electronics Techn--logy, No12,1998

[3] Lu xueping . “The key technology of Interactive Television”,Jounal of Television technology ,No3,1998

[4] “VOD Technology and Appplication on The HFC network”, Broadcasting Technology ,1998

[5] Han Runtan ,Tian liyan. “VOD Implement System”, 1998

Segmentation of Range Image based on Mathematical Morphology

Tao Hongjiu
Wuhan University of Technology
Wuhan, Hubei, 430063, China
E-mail: thjll@263.net taohongjiu@sohu.com

ABSTRACT

This paper presented a method of range image segmentation based on Mathematical Morphological. At first, the edges are extracted by morphological edge detector and threshold edge map, 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 normals 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 show 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:

$$g^x(x) = g(-x)$$

the grayscale dilation and erosion of a function f(x) by g(x) are defined by

$$[f \oplus g^s](x) = \max_{x \in D, z-x \in G} \{f(z) + g(z-x)\} \quad (1)$$

$$[f \Theta g^s](x) = \max_{x \in D, z-x \in G} \{f(z) - g(z-x)\} \quad (2)$$

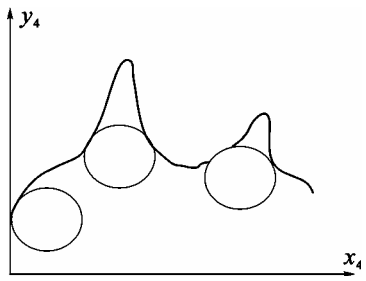
where D is the domain of f(x).

Grayscale opening and closing is another set of dual operations:

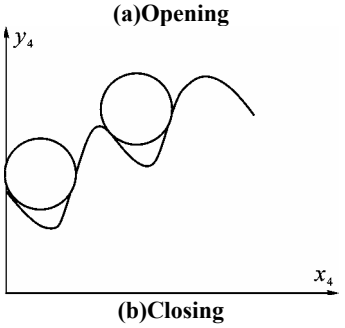
$$f_g(x) = [(f \Theta g^s) \oplus g](x) = [f(x) \Theta g(-x)] \oplus g(x) \quad (3)$$

$$f^g(x) = [(f \oplus g^s) \Theta g](x) = [f(x) \oplus g(-x)] \Theta g(x) \quad (4)$$

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, they are enhanced by rolling ball transformation. 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 edge orientation is determined by the shape of structuring element g. 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 the entire topography is flooded from below by letting water rise through the holes at a uniform rate. When the rising water in distinct catchment 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 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.

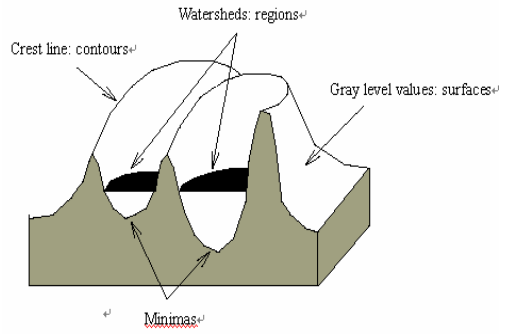


Figure 2 Gray-tone Valley segmentation algorithm

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 recursivel y, 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 minimas of edge map, get the markers;
d. Distance transform and get the catchment basins;
e. Extract crest line by valley segmentation.

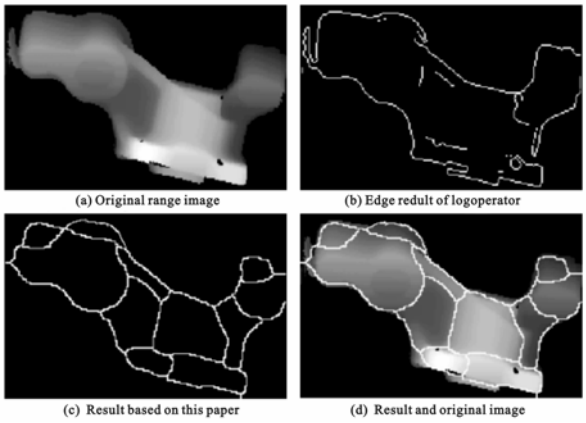


Figure 3 Experiment result

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 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.

5. REFERENCE

[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

Tao Hongjiu received the B.S. degree in computer and information science from Tianjing Industry University, Tianjing, China, in 1984, and doctor candidate of Institute of Pattern Recognition and Artifical Intelligence, HuaZhong University of Science and Technology, Wuhan, China, in 2000. He is a associate professor in the Department of Mechanistic Engineering, Wuhan University of Technology, China. His research interests include remote senseing image processing, digital watermarking, artificial intelligence and computer vision etc.

Applying Accountability and Atomicity to Secure Electronic Transaction

Bo Meng
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail: mengbo@263.net.cn
Qianxing Xiong
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail: qxxi@public.wh.hb.cn
Xinming Tan
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail: tanxming@public.wh.hb.cn

ABSTRACT

In electronic commerce protocols, atomicity and accountability are very important properties. Atomicity consists of money atomicity, goods atomicity and certified delivery, while accountability consists of money accountability and goods accountability. If an electronic commerce protocol has not these properties, the electronic commerce transactions under this protocol are prone to dispute and error. Secure Electronic Transaction (SET) is widely used electronic commerce in the world by credit card. But it neither is a protocol of goods atomicity and certified delivery nor has the properties of money accountability and goods accountability. In this paper, we propose an improved version of SET which has the properties of goods atomicity, certified delivery, money accountability and goods accountability for not only digital goods but also non-digital goods. And we analyze those properties of the improved version of SET..

Keywords: electronic commerce, atomicity, accountability, Secure Electronic Transaction.

1. INTRODUCTION

With the development of electronic commerce, many electronic commerce protocols have been suggested. The common properties of electronic commerce protocols include message integrity, privacy, non-repudiation [2], [3], [4], atomicity [5], [17], [18], causality [6], accountability [7], [8], [9], [10], [11], [12], [13] etc. Among those properties, atomicity and accountability are very important to electronic commerce protocols since they can improved its ability handling the disputation of payment and goods delivery in electronic commerce.

J.D.Tygar defines three levels of atomicity to electronic commerce protocols [5]: money atomicity, goods atomicity, and certified delivery. Money atomic protocols affect the transfer of funds from one party to another without the possibility of the creation or destruction of money. This is a basic level of atomicity that each electronic commerce protocol should satisfy. For example, Secure Electronic Transaction [1] (SET), which was developed by MasterCard and Visa corporations, is used widely in the world by credit card, is money atomicity protocol; But Digicash [15] is not money atomic protocol. Goods atomic protocols are money atomic, and also affect an exact transfer of goods for money. That is, if one buys a good using a goods atomic protocol, one will receive the good if and only if the money is transferred. For network protocols there must be no possibility that one can pay without getting the goods, or get the goods without paying. For example, SET is not goods atomic protocol. Certified delivery protocols are money atomic and goods atomic protocols that also allow both a merchant and a customer to prove exactly which goods were delivered. For example, NetBill [16] protocol is certified delivery protocol.

Accountability in electronic commerce protocols is the property whereby the association of a unique originator with an object or action can be proved to a third party (i.e., a party who is different from the originator and the prover), which concerned with the ability to show that particular parties who engage in such protocols are responsible for some transactions [7]. In particular, the accountability involves the ability of a party, called a prover, to convince to another party, called a verifier. Traditionally, the accountability is used only to resolve disputes among parties. In such cases, a judge would act as the verifier, and a defendant would act as a prover.

In the practical aspects the accountability consists of money or payment accountability [10] and goods accountability [10]. In particular, the money accountability is about the authorized transfer of money from customer's account to merchant's account. Goods accountability is about the authorized order of goods by a customer. The goods accountability can be used to resolve disputes on the mismatch between the goods that is ordered by a customer and the goods that is delivered by a merchant. And the goods accountability can be also used to deal with the goods atomicity and the certified delivery of electronic commerce protocol. We think that the goods accountability should not concerned with the price of the goods but the goods itself.

According to the definition of atomicity and the SET specification, we know that SET is money atomic protocol. The financial network guarantees that SET is money atomic protocol. But goods atomicity and certified delivery is not addressed by SET.

E.V.Herreweghen in [9] analyzed SET on the special accountability called payment or money accountability by Supakom Kungpisdan in [10]. The analysis shows that SET lacks of payment or money accountability. Supakom Kungpisdan in [10] analyzed SET on the goods accountability

and showed it is lack of it. But they don't give the improved version of SET concretely and detailedly. Only do they give some recommendations.

In this paper, we present an improved version of SET, which is not only money atomic protocol but also goods atomic protocol and certified delivery protocol .At the same time it is of money accountability and goods accountability. In section 2, we describe the improved version of SET .In section 3 and 4,we analyze the protocol our introduced. Section 5 concludes our work.

2. IMPROVED VERSION OF SET

According to its specification, SET does not provide the accountability, goods atomicity, and certified delivery. In order to satisfy the practical needs, we introduce an improved version of SET, which provides the accountability, goods atomicity, and certified delivery. For the reason of simplicity, we do not consider additional SET options; such as separation of authorization .The improved version of SET is described as the following. In our improved version of SET, the steps we added are in the boxes.

$E_x(M)$: message M, encrypted under X's public key
$Signed_x(M)$: message M, signed with X's signature key (include M)
$S_x(M)$: message M, signed with X's signature key (does not include M)
$E_x(Signed_y(M, baggage))$: notation for $\{Sig_y(M,baggage),E_x(M),baggage\}$
TID_s: stands for transaction ID
HOD: H(HODContents)
HODContents: OD, PurchAmt, ODSalt
OD: Order Description
MID: Merchant ID
AuthX: X in Authreq
AuthResX: X in AuthRes
CapRatio: ratio of AuthAmt: PurchAmt
PI:
$S_c(HOIData,HPIData),E_A(PIHead,HOIData,PANData)$
HOIData: H (OIData)
HPIData: H (PIData)
OIData: TID_s, $Chall_C$, HOD, ODSalt, $Chall_M$
PIData: PIHead, PANData
PIHead: TID_s, HOD, PurchAmt, MID

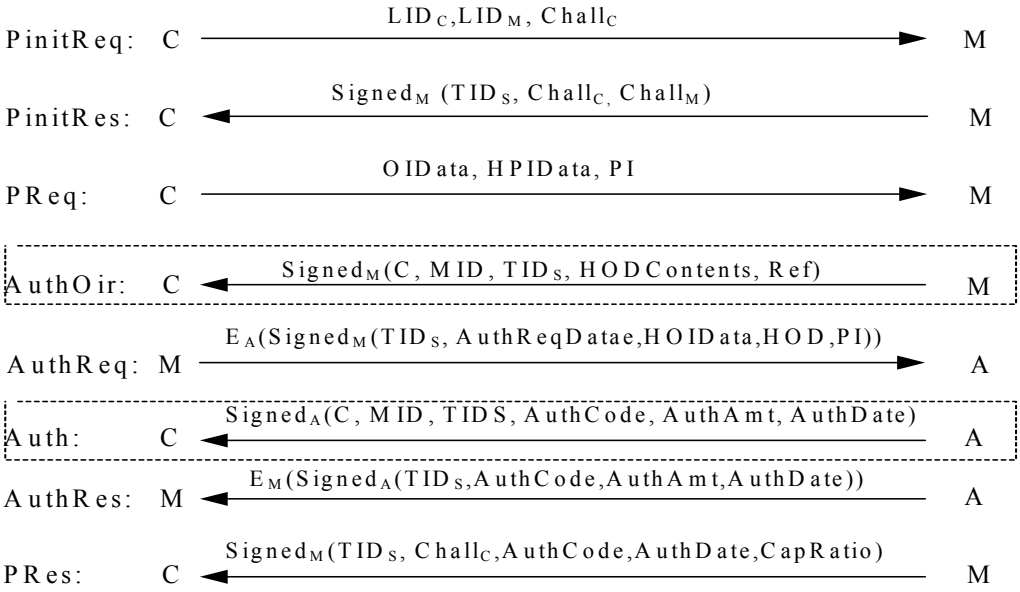


Fig. 1 the improved version of SET in-line authorization

We do not modify the structures of the messages of SET. Only two messages that are AuthOir and Auth are added in the improved version of SET, so that the legacy system can be used. The message AuthOir is used as the non-repudiable evidence that the merchant has received the message Preq and has agreed the HODContents the customer sent. The message Auth can be used as the non-repudiable evidence that acquirer has agreed to transfer money from customer's account to merchant's account.

3. ACCOUNTABILITY OF THE IMPROVED VERSION OF SET

First we analyze the money accountability of the improved version of SET, then analyze the goods accountability.

Analysis of Money Accountability

Three primitive transactions in SET are value subtraction, value claim, and payment. See Fig. 2

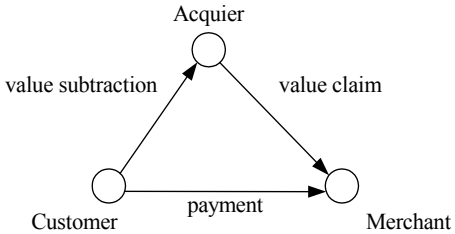


Fig. 2 primitive transactions

In value subtraction, customer allows acquirer to remove money from his account. In value claim, merchant requests the acquirer deposits money to his account. In payment, customer transfers money to merchant.

We can argue that the protocol is completely money accountability if it has non-repudiable proof of authorizations. From [9], the requirements of that proof are as the following:

M can prove "C authorized payment (C, M, Amount)", or Auth (C-M).
M can prove that C made a payment to him.
C can prove "M authorized payment (M, C, Amount)", or Auth (M-C). C can prove that M sent the receipt of payment to him.
A can prove "C authorized value subtraction (C, A, Amount)", or Auth (C-A). A can prove that C requested him to deduct money
from C'saccount
C can prove "A authorized value subtraction (A, C, Amount)", or Auth (A-C). C can prove that A sent the receipt of money deducted from C's account to him.
A can prove "M authorized value claim (M, A, Amount)", or Auth (M-A). A can prove that M asked the payment to be made to him.
M can prove "A authorized value claim (A, M, Amount)", or Auth (A-M). M can prove that A transferred (or committed to transfer) money to M's account.
After a proper execution of the improved version of SET, it will produce the evidence of customer, merchant and acquirer listed in Fig.3

Role	Evidence	Description
Customer	$Signed_A$(C, MID, TIDS, AuthCode, AuthAmt, AuthDate)	Auth(A-C)
	$Signed_M$(TID_S, $Chall_C$, AuthCode, AuthDate, CapRatio) $Signed_M$(C, MID, TID_S, HODContents, Ref)	Auth(M-C)
Merchant	E_M($Signed_A$(TID_S, AuthCode, AuthAmt, AuthDate))	Auth(A-M)
	$Signed_M$(C, MID, TID_S, HODContents, Ref) from Customer OIData, HPIData, PI	Auth(C-M)
Acquirer	E_A($Signed_M$(TID_S, AuthReqDatae, HOIData, HOD, PI))	Auth(M-A)
		Auth(C-A)

Fig.3 the evidences

According to our definition, we can prove that the improved version of SET is money accountability. Owing to the space limitations, we can't present the formal proofs with the logic framework suggested by Supakorn Kungpisdan in [10].

Analysis of Goods Accountability
We analyze the goods accountability of the improved version of SET with the logic framework suggested by Supakom Kungpisdan in [10] too. The definition of goods accountability in [10] is "goods accountability is about the authorized order of goods by a client. The goods accountability can be used to resolve disputes on the mismatch between the goods which is ordered and that which is delivered." According to the practical application and the definition of goods accountability, goods accountability should not concern the price or money, so we can use the evidence of Preq and AuthOir to prove the goods accountability. These evidences mean that customer and merchant had an agreement on goods. Customer can use it as evidence to solve the disputation when the merchant's behavior is fraud. At the same time, goods accountability guarantees the certified delivery according to the definition.
In the following, we prove the goods accountability of the improved version of SET.
The goal of proof:
M believes M CanProve (C authorized goods-order (C, M, OD, ref)) to V.
C believes C CanProve (M authorized goods-order (C, M, OD, ref)) to V.
First we discuss "M believes M CanProve (C authorized goods-order (C, M, OD, ref)) to V". In order to show that "M believes M CanProve (C authorized goods-order (C, M, OD, ref)) to V", it suffices to show that M believes M CanProve (C says (M, OD, Ref)) to V by providing the message of Preq to V. In the same way, we can prove "C believes C CanProve (M authorized goods-order (C, M, OD, ref)) to V." by providing the evidence of AuthOir.

4. ATOMICITY OF THE IMPROVED VERSION OF SET

Money atomicity is guaranteed by the financial networks because SET is only responsible for submitting money transfer requests and does not do the real money transfers. Our protocol achieves the goods atomicity and certified delivery by money accountability and goods accountability. If the merchant got the money but did not send the goods, the customer can provide the evidence of the message of Auth to solve the disputation. If the merchant sent the goods that are not ordered by the customer, the customer can use the evidence of message of AuthOir to solve the disputation.

5. CONCLISION

In this paper, we proposed an improved version of SET to over come the shortage of original SET on atomicity and accountability. We add the message of merchant authorized order AuthOir after the message of payment request PReq as the evidence of merchant authorized order Auth(M-C) and the message of acquirer authorized payment Auth after the message of merchant authorization request AuthReq as the evidence of acquirer authorized payment Auth(A-C). With these new introduced evidences and the original evidences provided by SET, payment and goods delivery disputes can be resolved. That is, our protocol has the properties of money atomicity, goods atomicity, certified delivery and money accountability and goods accountability for not only digital goods but also non-digital goods according to the definitions of atomicity and accountability. The improved version of SET

does not change the original message structures of SET, only adding two messages to SET, so the legacy system can be used continually. In the future, we will establish an electronic commerce website to test and evaluate our protocol.

6. REFERENCES

[1] MasterCard and Visa, SET protocol specifications, http://www.setco.org/

[2] ISO/IEC 13888.1 information technology - security techniques - non-repudiation - part 1: general, 1997.

[3] ISO/IEC 13888.2 information technology - security techniques - non-repudiation - part 2: mechanisms using symmetric techniques, 1997.

[4] ISO/IEC 13888.3 information technology - security techniques - non-repudiation - part 3: mechanisms using asymmetric techniques, 1997.

[5] J.D.Tygar, “Atomicity in Electronic Commerce”, CMU-CS-96-112, January 1996,School of Computer Science Carnegie Mellon University, Pittsburgh, PA 15213

[6] Douglas H.Steves, Chris Edmondson Yurkanan, Mohaned Gouda, “Properties of secure transaction protocols”, Computer Networks and ISDN Systems 29(1997) 1809-1821

[7] R.Kailar. “Accountability in electronic commerce protocols”. IEEE Transaction on Software Engineering, 1996

[8] R.Kailar, “Reasoning about accountability in protocols for electronic commerce”, In Proceedings of the IEEE Symposium on Research in Security and Privacy, Oakland, CA, May 1995.

[9] E.V.Herreweghen, “Non-Repudiation in SET: Open Issues”, In the Proceedings of the Financial Cryptography, 1999

[10] Supakom Kungpisdan, yongyuth Permpoontanalarp, “Practical Reasoning about Accountability in Electronic Commerce Protocols”, In Proceeding of the 4th International Conference on Information Security and Cryptology, Seoul, South Korea, 2001, Lecture Notes in Computer Science, Springer Verlag

[11] S. Kungpisdan and Yongyuth Permpoontanalarp, “A Logic for Solving Disputes in Electronic Commerce”,In Proceedings of 24th Electrical Engineering Conference (EECON24), 2001

[12] E.V.Herreweghen, “Using Digital signature as Evidence of Authorizations in Electronic Credit Card Payments”, Research report 3156,IBM Research, 1999

[13] N.Asokan, E.V.Herreweghen, M.Steiner, “Towards a Framework for Handling Disputes in Payment Systems”, In the Proceedings of the 3rd USENIX workshop on Electronic Commerce, Boston, Massachusetts, 1998

[14] V.Kessler, H.Neumann, “A Sound Logic for Analyzing Electronic Commerce Protocols”, ESORICS’98 Proceedings of the fifth European Symposium on Research in Computer Security, 1998

[15] http://www.digitcash.com

[16] A.Somogyi, T.Wagner, et al, “NetBill, Information Networking Institute Technical Report TR 1994-11”, Fall, 1994

[17] Jiawen Su, J. D. Tygar, “Building Blocks for Atomicity in Electronic Commerce”, in the Proceedings of the Sixth USENIX UNIX Security Symposium San Jose, California, July 1996.

[18] Jean Camp, Michael Harkavy, J.D.Tygar, Bennet Yee, “Anonymous Atomic Transactions”, CMU-CS-96-156, July 1996

LogP Analysis of User Level Network Cluster System *

Zhihui Du, Yu Chen, Ziyu Zhu, Haofei Liu, Chao Xie, Sanli Li
Department of Computer Science and Technology
Tsinghua University,100084, Beijing, China
E-mail: duzh@tsinghua.edu.cn

ABSTRACT

LogP is a distributed parallel computation performance model suitable for cluster system. In this paper, the LogP model is used to assess and analyze a cluster system--THNPSC-II's two user level networks: VIA(Virtual Interface Architecture) and GM. VIA is the industry standard of user level network communication. GM, which is the communication software system supported by Myrinet hardware products, can provide user level network APIs for high level applications. In this paper, the VIA performance result is provided by THVIA which is a VIA implementation based on hardware support instead of the commonly software or firmware simulation. To compare with traditional kernel level network, such as TCP/IP protocol network, the second LogP test based on MPI APIs is also provided. Although the kernel level network LogP test is run on the same hardware as user level network , the user level network can provide much better performance. At last the third LogP test on user level network but with high CPU payload indicates that CPU contention during message processing can decrease the communication performance significantly. Based on the test results, multiple viewpoints analysis is employed to assess the user level network communication. The primary VIA supported cluster system THNPSC-II has shown that user level network is critical for a high performance cluster system.

Keywords: LogP Model, User Level Network, Virtual Interface Architecture, Cluster System.

1. IMPORTANT

Cluster computing [2] is becoming more and more important and popular. High performance network is a key component of cluster system, not only the network hardware, but also, even more important, the network support software. Now a new kind of communication style ---ULN(User Level Network) Communication[3] has been employed into cluster system widely. ULN communication aims to provide low latency , high bandwidth and high reliable communication for SAN(System/Server/Storage Area Network)or cluster system. To achieve this object, ULN eliminates the OS from the communication critical path compared with the KLN(Kernel Level Network). This means that user level applications can exchange messages by directly accessing the high performance hardware communication device without the involvement of OS. So the overhead of OS intervention in legacy protocol implementation, such as memory copy, interruption and context switch, can be eliminated.

VIA(Virtual Interface Architecture)[9] is the industry standard for ULN communication. Intel, Compaq and Microsoft are three jointed authors of VIA. Hundreds of important academic and industry organizations also contribute to the VIA specification. VIA is composed of four parts,VI, VI provider, VI consumer, and CQ(Completion Queue). VI is the endpoint which provides connection oriented communication between two applications. VI consumer uses VI to transmit message with the support of VI provider. CQ is an optional part of VIA. All completed communication descriptors can be pushed into this queue. User applications poll the CQ to find if a communication descriptor has completed.

THVIA [18]is a VIA system which has fully passed Intel's early adopter test and most of functional and conformance tests, at the same time, we port Intel's performance test suits Vipperf [8]to Linux and the THVIA system can compile and pass all of Vipperf test programs. With the help of dedicated VIA supported hardware network THNet, THVIA can provide all necessary VI functions needed by the standard VIA specification. THVIA is designed to be compatible with TCP/IP protocol considering the current and former applications. Myrinet [14] is a kind of high performance SAN. It can also provide user level network communication via its software package GM. But GM is another kind of user level communication protocol which can not be compatible with VIA. Now Myrinet are extending GM package to support VIA specification. Both THVIA and GM can support TCP/IP protocol. This paper gives and analyzes the test results over THVIA and GM based on LogP model.

2. TEST DESIGN

LogP model

LogP model [4][5] is provided to assess network communication performance and predict application performance. This model uses four parameters to measure the performance of a given network.

- L(Latency) : the time spent on network to transmit a message
- o(overhead) : the time spent on sender and receiver sides to send or receive a message
- g(gap) : the minimum time interval between consecutive messages
- P(Processors) :the number of processors

LogP test is based on point to point communication. All tests are run between two processors. One processor sends messages and the other receives them. To get the four parameters, this model needs two steps. The first step measures the Round-Trip-Time (RTT). The sending processor sends one message, the receiver receives it and (without changing the contents) sends it back to the sender. The total time of a message from the sender to receiver and back to sender is RTT time. The second step required for the LogP model is for the sender to send messages continuously, with the receiver acking and the sender handling incoming messages (the acks) as well as sends. This test is used to create LogP curves and from which the LogP parameters can be calculated.

The total software overhead o can be divided into sender overhead os and receiver overhead or,o=os+or. Let M be the maximal messages number that the sender can push into

* Research partly supported by HuaWei Technology Foundation and Tsinghua University 985 Foundation.

network continuously before the first ack message arrives, om be the time to send M messages, then os=om/M. Let Delta be the delay time between two continuous messages. Let T be the average time that the sender need to send one message, delay Delta, and deal with an ack from receiver. If Delta is bigger than the message transmission time, then or= T-os-Delta and L=RTT/2-os-or. If Delta is less than the message transmission time, then g=T.

Test program design

In this paper,the LogP test program in GM package is used to test Myrinet's performance. At the same time, we modify GM's LogP test program to make it suitable for THVIA test. The LogP test program provides by GM has so many goto sentences and control switch flags. So we modify the structure in our LogP program for VIA test. We separate the test into two parts from structure clearly, first get the RTT(this part just likes a pingpong test program), then burst send messages to generate the LogP curves. The THVIA LogP test program uses VIA's API but keeps the same usage and output file format as GM's LogP program. This makes the comparison easily. At the sender side, the command is

logp_test -s -o -t -h recvhostname -q -q -q

At the receiver side, the command is

logp_test -r -o -t -h sendhostname -q -q -q

After it finishes, the sender machine can get the test result files in directory logp_res/.

In order to compare with the kernel level network, we also develop a LogP test program using MPI's APIs[13]. This program need two processes. The process 0 as the sender and process 1 as the receiver. To satisfy the LogP semantic, the sender side must poll to find if send operation has finished and if there is incoming message. We use MPI_TEST and MPI_IPROBE no-blocking subroutines to implement the polling function. The output file format is the same as the ULN test. For the MPI LogP test , the command is

mpirun -np 2 logp_test -o -t -q -q -q

To analyze the impact of CPU contention during communication, we design the third test. When the LogP test is going, issues other task to generate CPU or network contention(in our test, we simply start a compilation task). From this test we can see how the CPU contention affects the communication performance.

Totally we design three kinds of tests. LogP test on ULN, LogP test on KLN but over the same hardware network as ULN and LogP test on contention ULN.

Test environment

The cluster system used to LogP test is named THNPSC-II. It consists of 16 computing nodes. Eight nodes are Intel's dual Pentium III 500 and other eight nodes are Intel's dual Pentium 733 processors. THNPSC-II has three kinds of hardware networks. The first is 100M Ethernet network. The second is Myrinet which consists of Myrinet-2000 Switch Enclosures M3-E64 and Myrinet-2000-SAN/PCI interface PCI short card M3M-PCI64C-2. The third is THNet, which is developed by our research group. THNet consists of THNC(32bit,66/33Mhz network interface adapter) and THSW(crossbar switch). The OS is Linux with kernel version 2.2.x. The motherboard of THNPSC-II cluster system only has 32 bit 33Mhz PCI slots.

3. TEST RESULTS ANALYSIS

Based on the LogP test results, we can provide three kinds of analysis. The first is performance analysis, the second is curve smoothness analysis, and the third is contention analysis. Because of the limitation of pages, we focus on THVIA ULN analysis and put some test results of ULN Myrinet in appendix part for reader's further reference.

Performance analysis

In Figure 1, (a) is ULN LogP test over THNet,(b) is KLN LogP test over THNet. The ULN bandwidth of THNet increases very fast at beginning(length less than 4KBytes), then slow, at last, almost no increment. Figure 7 is the corresponding bandwidth curves of Myrinet just like THNet. From Figure 1 we can find that the KLN's peak bandwidth is only about 22% of ULN's peak bandwidth over the same THNet hardware network. For Myrinet, the result is similar, The KLN's peak bandwidth is about 33% of ULN's peak bandwidth. This test result indicates that most of the hardware network bandwidth can not be used if KLN communication is employed, so ULN communication is very necessary to achieve high performance communication in cluster system.

We borrow the N1/2 concept in supercomputer Linpack Test and provide the HBWL concept. Like N1/2 which means the problem size for achieving half of maximal performance, HBWL(Half BandWidth Length) is the message length for achieving half of peak bandwidth. The sustained peak bandwidth of ULN THNet is about 106 MBytes/s, to achieve half sustained peak bandwidth the HBWL is about 1KBytes. Based on the research result of[7], 80% messages are small message(length less than 1KBytes), only 8% messages are bigger than 8KBytes. this means that ULN THNet can provide high bandwidth for most of the message. The same conclusion is suitable for ULN Myrinet. The sustained peak bandwidth of ULN Myrinet is about 75Mbyte/s, and HBWL of ULN Myrinet is 750bytes. For KLN THNet or Myrinet, the HBWL is much large. This means that not only the sustained bandwidth of KLN is very limited, but also the peak bandwidth is not easy to achieve for most messages.

Figure 2 includes the RTT, gap and Latency curves of LogP test over THNet. (a) is the ULN curve, (b) is KLN curve. Figure 8 is the corresponding curves over Myrinet. The ULN RTT over THNet is about 22 microsecond for 8 bytes message, but for KLN, the corresponding RTT is 110 microsecond. The performance of many applications is very sensitive to short message latency, so the RTT results also indicate that the ULN is superior to KLN. The bandwidth curves of Figure 1 and 7 are calculated from the gap curves of Figure 2 and 8. The RTT should be linear with message length, just as the THNet and Myrinet's ULN RTT curves show. But the KLN RTT curves of THNet and Myrinet is not so. The reason is that in the calculation of Latency, the time cost of MPI_TEST and MPI_IPROBE subroutines are taken as the send overhead. We are trying to improve our MPI LogP test program to avoid introduce additional cost into LogP test. Although there is also additional overhead in ULN LogP test program, it can be ignored. In KLN test, because the the polling subroutines are implemented by OS, the time cost will be much big and it can not be ignored.

Figure 6 is the original send burst curves and the RTT, Latency and gap curves are calculated from it. (a) is ULN curve and (b) is KLN curve. By comparing the two kinds of curves we can find that the message passing time for KLN is much

longer than ULN. It's shape is just like the LogP model's theoretical curves.
Table 1 gives the comparison of THNet and Myrinet's RTT, bandwidth and HBWL parameters. The ULN RTT time of THNet and Myrinet for message length 8 bytes is about 20us, The ULN bandwidth of THNet and Myrinet for message length 16000 bytes is about 100 MBytes/second. Both ULN THNet and Myrinet can arrive or overcome its half peak bandwidth when the message length reaches 1KBytes .

Smoothness analysis
In this analysis we provide another viewpoint to assess the performance of ULN communication, it is smoothness. In Figure 1 the trend of bandwidth curve is increasing, but it decreases at some points. This is because the THVIA implementation splits big messages into 4K small frames, the disassembling and assembling operations will decrease the bandwidth in some degree. So the bandwidth curve is not smooth at points of length about 4K, 8K, 12K and 16K. For ULN Myrinet bandwidth curve, it is very alike.
We also find that the sharp points not only appear at disassembly points, but also at other points(Figure 1). This is because of CPU or network contention. If CPU is switched to other task during the message processing stage, this will decrease the bandwidth. The sender and the receiver both push messages into the network can also decrease the available bandwidth.
The latency curve of (a) in Figure 2 has some irregular change. The test is repeated many times and this sharp points appear with high probability. It's because in our LogP test, the RTT test is passed first, then the send burst test to generate the LogP send burst curves which is independent with the RTT test. The send overhead can be affected greatly by system environment, so big send overhead can occur in many cases, such as CPU schedule, network contention. If this occurs, the latency, which is calculated by RTT-os-Delta, will be very small. In the KLN tests, we also find that the Latency curves tend to down which is also for the big os. But in KLN, big os is mainly because of the MPI_TEST and MPI_IPROBE 's cost.
To check the fine change of different curves, we do other tests with more sampling points. Figure 3 and Figure 4 are the curves corresponding with Figure 1 and 2 to show the fine changes of different curves. When more sampling points are tested, we can find the small change of LogP curves. Smoothness reflects if the communication performance can be provided steadily. If the LogP curves are too easy to affect by other factors, it means the available communication performance may change greatly. So some methods should be chosen to make the curves smoother.

Contention analysis
In this part we analyze the communication performance in contention environment. The LogP test should be done when the network and CPU are exclusive, but in real applications these conditions can not be gotten easily. So we test the ULN performance when CPU is heavily load or the network is high traffic. Both CPU payload and communication traffic may decrease the ULN communication performance. For simple, we just test the CPU payload. In fact, the effect of CPU payload and network traffic is same so these two effects can be exchanged in some manner.
Figure 5 and Figure 9 give the LogP curves when other tasks are added to test nodes during the LogP test. Although the trend of the curves is same as Figure 1 and Figure 2, there are so many sharp points in the contention condition curves. In the viewpoint of smoothness analysis, the smoothness of the curves in Figure 5 and Figure 9 is very bad. It means CPU contention can greatly decrease the ULN performance, so in real applications CPU or network contentions should be avoid as much as possible.
Although ULN can provide higher communication performance than KLN, to fully use this performance, the user applications and ULN communication support environment should be careful designed to avoid CPU or network contention. Overlapping communication with computation is an important method to improve real application's performance.

4. RELATED WORKS

Many projects are on ULN protocol, such as SHRIMP[12],U-net[17]. When the standard of ULN --VIA occurs, ULN research focuses on VIA specification and VIA implementation. Intel[8], NERSC'S M-VIA[15], Berkeley's B-VIA[16], Tsinghua[18] and IBM[10] all have their VIA implementation. These research focuses on design or implementation ULN or VIA protocol. After LogP model[5] is provided, [4]gives how to assess fast network. [6]use LogP model to assess application performance on CM5 machine. [1]adds a new parameter G to capture the large message bandwidth. There are other kinds of extension based on LogP model. This paper uses LogP mode as a tool to assess the ULN performance and compare with KLN performance. Besides the bandwidth and latency performance analysis, we employ smoothness analysis and contention analysis on ULN to give other information of ULN communication. From these analysis, this paper not only tell us what the performance of ULN communication is, but also how to improve the ULN performance and how to avoid the performance decrease of ULN performance.

5. CONCLUSION

In this paper we use the LogP model to analyze the VIA ULN performance. Performance analysis uses bandwidth and latency as the main assessing parameters. Smoothness analysis uses the curves' smoothness as another assessing parameter. Smoothness can give us more information which can reinforce the performance analysis. In general the communication test are CPU and network exclusive, but this is not the real adaptive condition for most applications, so we design the contention analysis. This analysis tells us how to achieve high performance in real contention environment.
From the LogP test results on ULN and KLN, the benefit of ULN is obvious. So we conclude that high performance cluster system must adopt ULN to improve its performance. Without ULN, cluster applications can not get even half of the hardware network performance. A cluster system connected by hardware support user level network can provide high performance close to the traditional supercomputer or MPP (Massively Parallel Processors) systems. Cluster computing is promising and ULN will help cluster computing to be more powerful.

6. REFERENCES

[1] A. Alexandrov, M. Ionescu, K.E. Schauser, and C. Scheiman. LogGP: Incorporating Long Messages into the LogP Model–One Step Closer Towards a Realistic Model of Parallel Computation. Proc. Seventh Ann. ACM Symp. Parallel Algorithms and Architectures,

pages 95–105, July 1995.
[2] Anderson, T.E.; Culler, D.E.; Patterson, D. A case for now (networks of workstations). IEEE Micro, 15(1):54–64, Feb. 1995.
[3] Bhoedjang, R.A.F.; Ruhl, T.; Bal, H.E. User-level network interface protocols. Computer, 31(11):53–60, Nov. 1998.
[4] Culler, D.E.; Lok Tin Liu; Martin, R.P.; Yoshikawa, C.O. Assessing fast network interfaces. IEEE Micro, 16(1):35–43, Feb. 1996.
[5] D.E.Culler et al. LogP:Towards a realistic model of parallel computation. Proc. Fourth ACM SIGPLAN Symp. Princi-ples and pratice of parallel programming, pages 1–12, 1993. New York.
[6] Dusseau, A.C.; Culler, D.E.; Schauser, K.E.; Martin, R.P. Fast parallel sorting under LogP: experience with the CM-5. IEEE Transactions on Parallel and Distributed Systems, 7(8):791–805, Aug. 1996.
[7] Gusella, R. A measurement study of diskless workstation traffic on an ethernet. IEEE Transactions on Communications, 38(9):1557–1568, Sep. 1990.
[8] Intel. Intel Virtual Interface (VI) Architecture Developer's Guide. http://developer.intel.com/design/servers/vi/developer /ia_imp _guide.htm, Sep. 1998.
[9] Intel,Compaq,Microsoft. Virtual interface architecture specification. Technical report, Dec. 1997. http://www.viarch.org/.
[10] M. Bazikazemi, V. Moorthy, L. Herger, D. K. Panda, and B. Abali. Efficient Virtual Interface Architecture

[11] Support for IBM SP Switch-Connected NT Clusters. In Proceedings of International Parallel and Distributed Processing Symposium, pages 33–42, May 2000.
[12] M. Blumrich, C.Dubnichi, E. W. Felten, and K. Li. Virtual memory mapped network interfaces. IEEE Micro, 15(1):21–28, Feb. 1995.
[13] Marc Snir,et al. MPI–the complete reference. MIT, second edition, 1998. http://www.mpi-forum.org/.
[14] Nanette J.Boden, Danny Cohen, Robert E.Felderman, Alan E.Kulawik, Charies L.Seitz,Jakov N.Seizovic, Wen-King Su. Myrinet: A gigabit-per-second local area network. IEEE Micro, 15(1):29–36, Feb 1995. http://www.myri.com.
[15] National Energy Research Scientific Computing Center. M-via: A high performance modular via for linux. 1999. http://www.nersc.gov/research/FTG/via/index.html.
[16] Philip Buonadonna, Andrew Geweke, David Culler. An implementation and analysis of the virtual interface architecture. In Proceedings of Superomputing, pages 7–13, Nov. 1998. http://www.cs.berkeley.edu/ philipb/via/.
[17] 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, pages 40–53, Dec. 1995.
[18] Zhihui Du, Liantao Mai, Ziyu Zhu, Haofei Liu, Ruichun Tang and Sanli Li. Hardware Based TH-VIA User Level Communication System Supporting Linux Cluster Connected by Gigabit THNet. 2001 International Symposium on Distributed Computing and Applications to Business, Engineering and Sci-ence (DCABES 2001), pages 134–138, October 2001. http://hpclab.cs.tsinghua.edu.cn/ duzh/THVIA.htm.

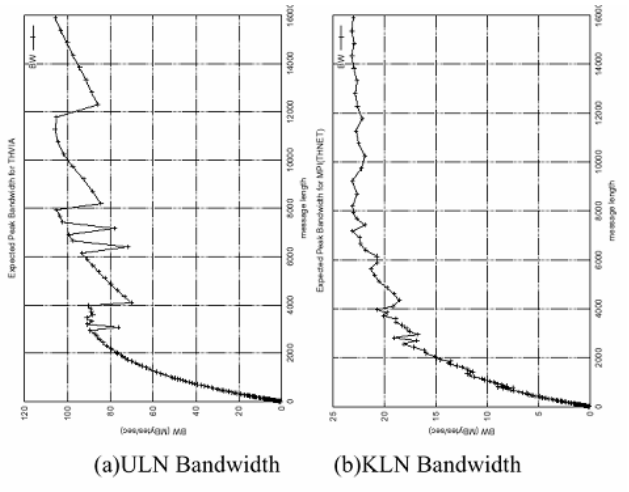


Figure 1 bandwidth curves of THNet

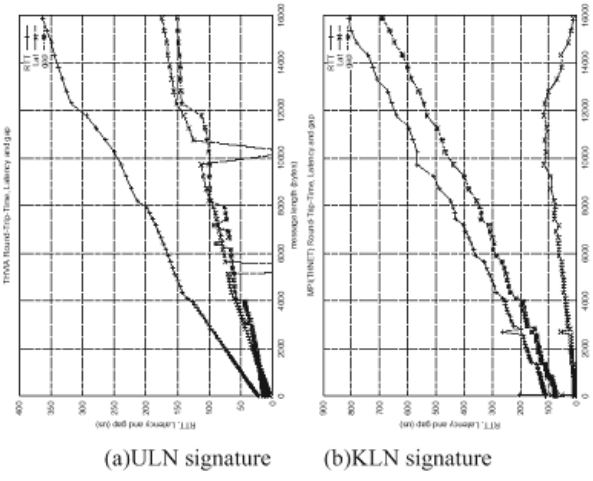


Figure 2 RTT,gap and Latency curves of THNet

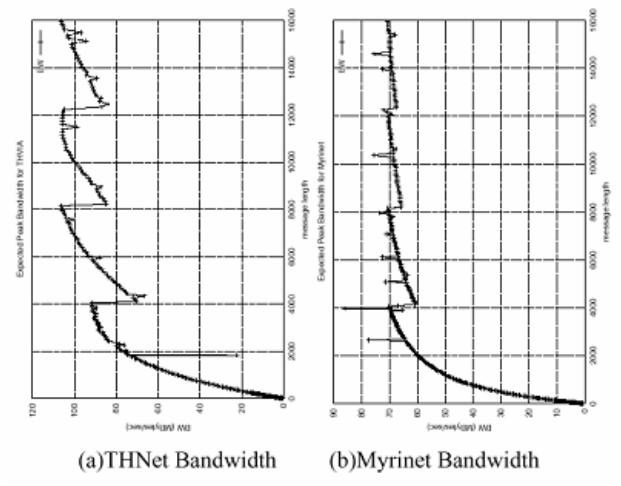


Figure 3 Fine Scale Bandwidth curves of THNet and Myrinet

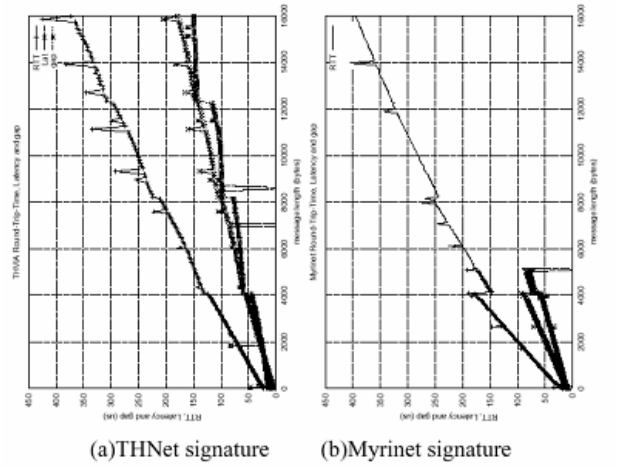


Figure 4 Fine scale RTT,gap and Latency curves of THNet and Myrinet

Table 1 Comparison of THNet and Myrinet

Parameters	THNet		Myrinet	
	ULN	KLN	ULN	KLN
RTT(us)	22	110	18	140
BW(MByte/s)	106	24	75	25
HBWL(Bytes)	1000	1500	750	2100

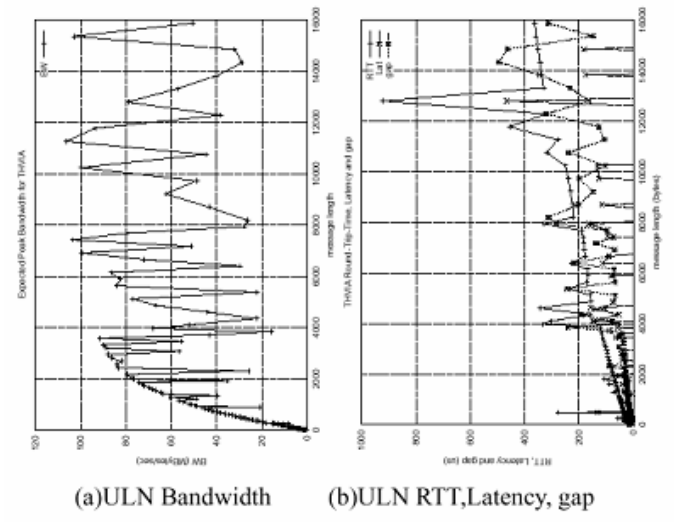


Figure 5 LogPcurves on THNet when CPU payload

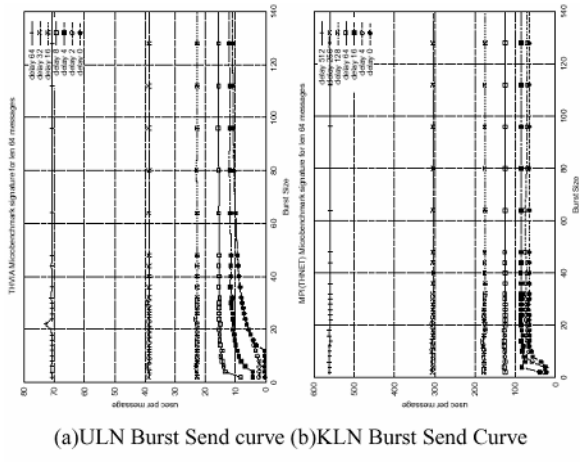
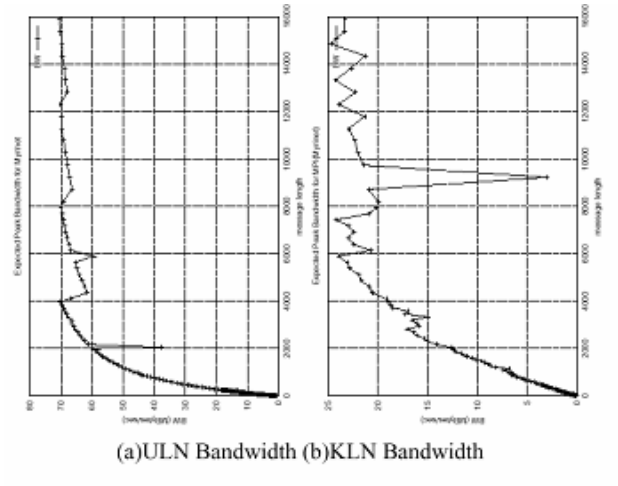
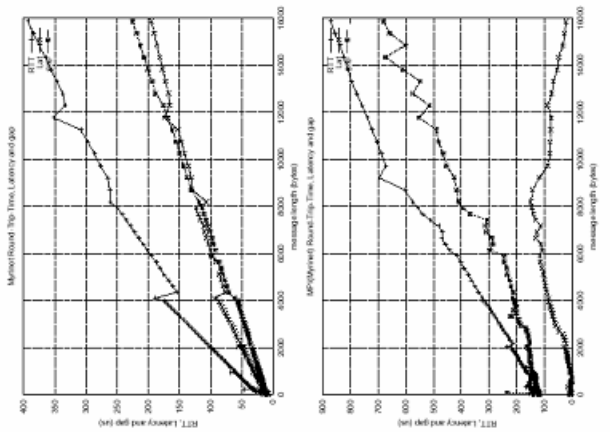


Figure 6 LogP Send Burst curves of THNet

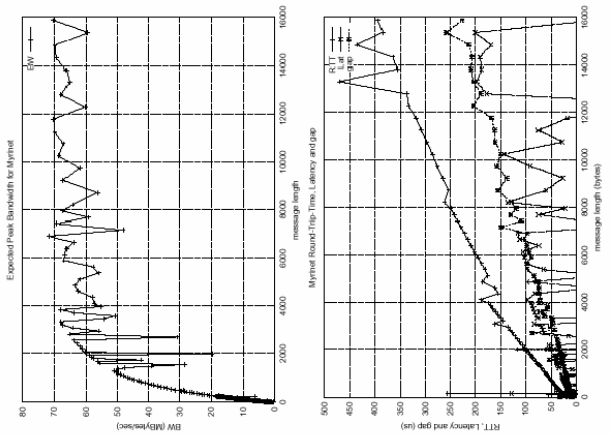


Fingue7 Bandwidth curves of Myrinet



(a)ULN signature (b)KLN signature

Figure 8 RTT, gap and Latency curves of Myrinet



(a)ULN Bandwidth (b)ULN RTT,Latency, gap

Figure 9 LogP curves on Myrinet when CPU payload

Study on Resource Sharing and Interlinkge of Computer Different Network*

Zhu Jinjun , Yang Kuihe , Zhang Xiaoming, Zhang Xuemei
College of Information, Hebei University of Science and Technology, Shijiazhuang Hebei China
E-mail: ykh@hebust.edu.cn

ABSTRACT

Along with the development of computer network technique, many kinds of different network and network operating system exist together. Solving resource sharing and interlinkge of computer different network is becoming more and more important. The studied basic idea of different network interlinkge realizes linking of different computer network and solves the problem of resource sharing and interlinkge of computer different network.

Keywords: network, linking each other, operating system, sharing

1. INTRODUCING

During the development of computer network technology, many kinds of network have emerged. The heterogeneousness of these network is in two aspects: one is the of the low level network technology for example Ethernet, ATM and FDDI; the other is the heterogeneousness of the network operating system for example Unix, NetWare, Windows NT, and Linux. Along with the widely application of computer network, how to interlink these heterogeneous networks to share network resources is a problem to solve urgently. In many application areas, real-time communication is emphasized, which need real-time collection, processing and controlling of data. So research on resources sharing and interlinkage of heterogeneous networks is very significative.

2. FUNDAMENTAL IDEA OF INTERLINKAGE OF HETEROGENEOUS NETWORK

TCP/IP preferably solved the problem of the interlinkage on internet. Some mainstream network operating system also provided gateway software to transform between protocols, which can realize the access of heterogeneous network. But the interlinkage and access through these manners have some limitations and their destination is resource sharing for example file service, printing service etc. All of them didn't provide direct real-time communication of heterogeneous network. However, in the industry control area, the real-time of communication is usually emphasized, which need real-time collection, processing and controlling of data. If data is transferred through method of resource sharing, the transfer efficiency will be greatly depressed and the real-time characteristic of data transfer can't be guaranteed. So we must research on the real-time communication of heterogeneous network. Heterogeneous computer network can be interlinked in some level, which can provide a single virtual network function interface to network application layer. This model greatly simplifies the realization of process communication of heterogeneous network. In order to express distinctly, the fundamental idea of the interlinkage of heterogeneous network is submitted. It can be divided into two independent layers, which are the linking on network level and the linking on application interface level

.

2.1 The Linking On Network Level

Heterogeneous networks' architecture and communication protocol are different. Facing these two aspects of heterogeneousness, the linking on network level is going to resolve the problems in order to make application localize on a unified network. The method to realize the linking on network level is to make use of TCP/IP .

For the low level heterogeneous network such as Ethernet and TokenRing, TCP/IP abstract and shield the detail of network hardware and provides an unified, cooperative and general communication system.

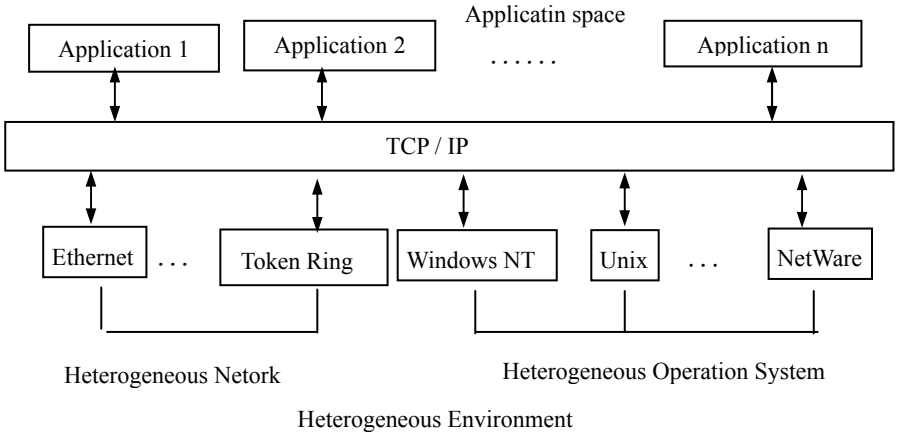


Fig.1 The linking on network level

Different network operating systems have different network communication protocol such as IPX/SPX, NetBEUI etc., but at present nearly all of mainstream operating systems support TCP/IP. So if the communication application based on different system all use TCP/IP, all networks are a kind of TCP/IP network to the programmers.

This idea can be expressed clearly by Fig.1 from which we can see that TCP/IP is between application space and heterogeneous environment space. All sorts of application directly communicate with TCP/IP and it is not necessary to care about what is below TCP/IP, that is to say, heterogeneous environment is transparent to programmers. TCP/IP shields downwards all kinds of heterogeneousness and provides upwards a unified interface to programmers. TCP/IP becomes a logical network on the network layer/transport layer, which can be looked as a platform. To the applications, this TCP/IP platform realized the linking on network level.

Therefore, the idea of the linking on network level is that no matter what system the application is based on, as long as all systems use TCP/IP as their communication protocol, in the view of programmers applications run on a unified TCP/IP network system and this system provide network communication supports to applications.

2.2 The linking on application interface level

After the linking on network level is realized, we own a unified TCP/IP network system. The final destination is the communication between applications. Application communicate with TCP/IP should depend on socket, so we can look sock as a middle interface layer. What is important is that different system's socket layer is different. The idea of the linking on application interface is to reduce this difference, namely, to realize the identical of application interface.

Socket is a interface on system-call level which is first submitted by UNIX and becomes the standard of the communication between process on UNIX network. Along with the application requirement, all kinds of mainstream operating system add socket interface successively, for example Winsock in Windows NT and PC/TCP socket API in DOS and so on. These criteria are extended more or less in order to fit different operating system kernel, which leads to some difference among these socket standards. The application working in heterogeneous environment can't ignore these differences. If a program on a system use an extended function by this operating system and a program working on the other system doesn't support this extended function, an error of process communication on network will occur. The socket standards are provided by different operating systems all inherit from UNIX Berkeley socket. Although there are differences among all kinds of socket standards, they have an intersection that is Berkeley socket. That is to say, all socket standards support Berkeley socket, therefore Berkeley socket fit all of the systems.

The main idea of the linking on application interface is no matter what operating system the application establishes on and no matter which kind of socket the application uses, as long as it uses the function of Berkeley socket, we can consider these applications establish on a unified socket interface layer and it can guarantee the correctly working of process communication on network.

The linking on network level and application interface level respectively realize the interlinkage of heterogeneous network in different level, which provide a approach to solve the problem of process communication on heterogeneous network. In the view of programmers, the heterogeneous network becomes a unified network. The process on heterogeneous network can be considered as a process on one network, and there is not necessary to think over the heterogeneousness. So the program design and realization are simplified to a great extent.

3. REAL-TIME COMMUNICATION ON HETEROGENEOUS NETWARE

3.1 Working Principle And Realization Method

From the application layer of OSI/RM, we can think the network to be constituted by client and server. But from the transport layer and network layer, there is no essential difference between client and server and they are both peer entities of network transfer, and the communication between them is so-called peer-to-peer communication. In peer-to-peer systems, an entity on a source-machine can communicate directly with the entity on a end-machine, and it seems that they communicate through a direct connection no matter how many middle nodes will be passed. There are three merits of peer-to-peer communication. Firstly, after the connection between source end and terminal end is established, once data send out from source end , the sender must know the receiver can receive the data; secondly, point-to-point data transfer doesn't need store and forward so that we can get higher efficient and shorter delay in the whole process of transfer; thirdly, point-to-point high level software doesn't need the function of store and forward which depend on the lower layer to realize store and forward, so that the design and realization of the high level software are very simple. Therefore, if the delay of network hardware is very short, point-to-point communication can guarantee the real-time characteristic of communication.

To realize the real-time communication of network application, not only development on lower layer is needed, but also the point-to-point communication should be adopted. The reason why we select protocol on lower layer is that the execution of protocol on higher layer will depend on the protocol on lower layer. The command from higher layer should be transformed to the function call on lower layer, so it is apparently slower than direct call. Different network operating systems usually have different choices. For example, IPX/SPX development interface on the third layer & the fourth layer usually are adopted in Novell NetWare system. However in heterogeneous network environment we have to develop network application with TCP/IP as we have discussed in the front. TCP/IP is a protocol cluster which include many protocols. On principle of developing on lower layer, as a matter of fact, only IP, TCP, UDP, etc. on the third and the fourth layer should be selected. To the second aspect, point-to-point communication can be established by adopting Client/Server model. The main idea of Client/Server model is that before the data transferring between the two processes on network, there should be a handshake process. That is to say, one side requests and the other side respond. When the handshake succeeds, a virtual circuit from source to terminal between the peer entities of two machines is established. Then data can transfer on this virtual circuit and we needn't care about whether the middle nodes exist. It is consistent with the manner of point-to-point communication, so we can develop real-time communication application by use of Client/Server model.

3.2 Time Sequence Cooperation Of System Call

The developments of real-time communication application on heterogeneous network usually adopt TCP/IP to link each other, and adopt Berkeley socket as developing interface, and adopt Client/Server model to form communication. Sychronization is a main problem of the realization of Client/Server model, which will depend on the time

sequence of system call. UNIX Berkeley socket interface provides system calls which are connection oriented or connectionless. Connection oriented socket mechanism can provide credible data transfer, and the typical time sequence of system call is as fig. 2.

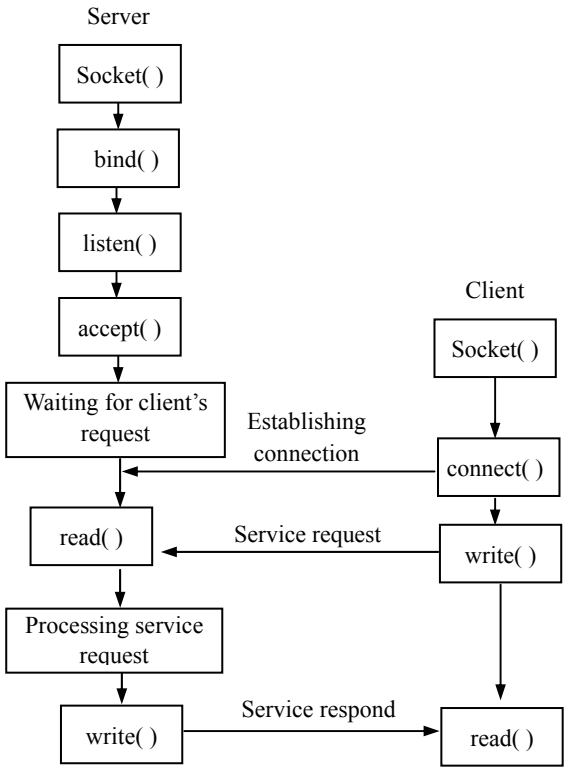


Fig.2 The system transferring time

4. CONCLUSION

This paper discussed the fundamental idea of the linkage of heterogeneous computer network, the realization method of real-time communication and the realization framework based on Client/Server model. By properly applying these theories, we can successfully realize resource sharing and real-time communication on heterogeneous network. The platforms include network operating systems involved in our project such as Windows NT, UNIX, NetWare, and Linux.

5. RENFERENCES

[1] Li Yong, Research of real-time communication based on heterogeneous networks, Wuhan: Journal of Wuhan Technical University of Surverying and mapping, Vol.24 NO.4 Dec., 1999, pp362-366.

[2] Wang Yongqun, The Interlinkage of Windows NT, Netware and Internet, Journal of ShanXi Finance and Economics University, Vol. 22 Supplement Dec., 2002, pp33-36.

[3] Li yang, Liu feng, Ji Biyou, The Inter working of Windows NT and NetWare, Microcomputer Development, Mar., 1999, pp38-40.

* This task is supported by Hebei province nature science fund (601259)

Congestion Control In Atm Networks Performances Of Becn Scheme Implemented In "C" And Analysis Of Kvs Algorithm

Santhana Krishnan SrinivasaGopalan
Department of Electronics and Communication Engineering, Kongu Engineering College,
Perundurai, Erode -Dt, TamilNadu Pin-638052, INDIA
E-mail: kris_elex@rediffmail.com
And
Vasanthan Balasubramanian
Department of Information Technology, Kongu Engineering College
Perundurai, Erode -Dt, TamilNadu Pin -638052, INDIA
E-mail: b_vasanthan@rediffmail.com
Under Guidance of Asst.Prof N.K.KarthiKeyan

ABSTRACT

Congestion in a packet switched networks is a condition in which performance degrades due to saturation of the network resources such as available communication links, processor speed and buffer capacity. Adverse effects resulting from such congestion are unacceptable delays, wastage of system resources and possibly total network failure. The problem of congestion becomes critical in Asynchronous Transfer Mode (ATM) networks due to diverse service requirements, high link speed and various characteristics of traffic. The ATM network has to provide guaranteed service to real-time applications such as voice and video. It accommodates non real- time application such as data with Available Bit Rate (ABR) services.

The Preventive congestion control scheme takes necessary action in restricting admission of calls to the network. In this the performances degrade for data application due to its bursty nature. In reactive scheme, dynamic regulation of cell emission rate of various sources is achieved using feed back from the networks. This is more suitable for bursty data services.

In this paper we developed an analytical model based on Discrete time Markov chain which can be applied for uniform and bursty traffic. The Peter Newman model has been implemented using C.We have Proposed a new idea of Modified BECN Scheme using our KVS algorithm.

Key Words: BECN, ATM, ABR

1. INTRODUCTION

1.1 ATM Networks

Broadband Integrated Services Digital Network (B-ISDN) was defined as a standard for transmitting voice, video and data simultaneously at speeds greater than 1.5 to 2Mbps. The technology selected to deliver B-ISDN was Asynchronous Transfer Mode. It uses statistical multiplexing technique [1]. The ATM has well defined standards and protocols for multi-service network infrastructure. Some of the features of ATM are:

1. The ability to provide very high speed asynchronous switching.
2. Combine the best properties of circuit switching (low delay and connection oriented type of networking) and packet switching techniques (statistical multiplexing)
3. Support for wide variety of applications ranging from the application requiring highly variable bandwidth to constant bandwidth.

1.2 ATM Traffic

An ATM network is a high-speed network where all information is segmented into a series of fixed-size short packets called "cells". The statistical multiplexing of cells belonging to various virtual connections will provide the users with the possibility of a bit rates varying in a wide range during connection based on demand. Together with the conventional Constant Bit Rate (CBR) and Variable Bit Rate (VBR) services supported in ATM networks, new service category called Available Bit Rate (ABR)[2]. The guaranteed forms a contract between traffic source and the network.

For CBR, ATM connection traffic will be described in terms of peak cell rate, cell delay variations. For VBR, ATM connection also adds sustainable cell rate and Burst Tolerance. For guaranteed service received by the network, the call acceptance control determines the resources that the call will required. If the required resources are available the call is accepted, if not it is rejected. Examples of traffic that may require a guaranteed service include real-time traffic such as circuit emulation, voice, and video.

However the existing data networks applications are incapable of preceding their own bandwidth requirements. Since the traffic characteristics are unknown an explicit guarantee of service can't be given. So a data application requires a service that dynamically shares the available bandwidth between all active sources. This service is referred to as best- effort service [3]. Recently it has been named as ABR service due to dynamic sharing of bandwidth available for this service between the users.

2. CONGESTION CONTROL SCHEMES IN ATM NETWORKS

The congestion control schemes used in ATM are classified into two categories namely Open Loop Control or Preventive Control and Closed Loop Control or Reactive Control [4].

2.1 Preventive Control

In preventive control the traffic parameters are first specified (peak bit rate, average bit rate). If there is available bandwidth then the network accepts the connection request and desired quality of service is guaranteed throughout the session. The preventive control takes necessary action to prevent congestion by restricting the admission of calls to the network. So the preventive control is basically an admission control mechanism. These congestion control schemes are applicable to CBR and VBR traffic. Examples of various preventive control mechanisms such as Leaky bucket and virtual leaky bucket mechanisms etc.,

2.2 Reactive Control

Closed loop control or reactive control schemes dynamically regulate the cell emission rates of the various sources by using feedback information from the network. This type of control takes necessary action to recover from the state of congestion and types are explained as follows:

2.2.1 Rate Based Approach
In this closed loop congestion control scheme if a queue length exceeds the threshold limit the source is made to slow down its rate of transmission by sending a feedback signal. This scheme was first proposed by peter Newman and called as Backward Explicit congestion notification (BECN) scheme.

2.2.2 Credit Based Approach
It works on a link-by-link basis. At the start of the transmission the receiver will allocate the buffer to the link depending upon the type of service the link can support. The receiver node sends the credit cell to the sender node. The credit cell contains the record of the amount of data cells forwarded or discarded by the receiver node [5]. The sender node has the record of the data cells, which it has transmitted. After receiving the credit cell the sender calculates the credit balance. The sender will keep on transmitting until the credit balance is greater than zero.

2.2.3 Integrated Approach
Rate based approach is the simplest approach and therefore it is easy to implement. The drawback of this method is that there can be buffer overflow in case of congestion and this can lead to cell loss. Rate based approach is unable to precisely allocate the resources in accordance with transient bursts. On the other hand there is no buffer overflow in credit-based approach and it can easily control transient bursts. But its drawback is that it requires large buffer memory at intermediate node. Hence it was suggested to use rate control in credit style, which is popularly called as Integrated Approach. However rate based approach is mostly used for closed loop congestion control method. Also its simplicity favors its suitability to Congestion control in ATM networks for ABR traffic.

3. PERFORMANCE MEASURES AND PARAMETER ANALYSIS

3.1 Backbone Link Analysis

We study the queuing behavior of a particular outgoing link in a backbone node using a discrete time Markov model where the time of the link is normalized by one cell transmission time. Let p_i (i =1, 2…L) denote the probability of i cell arrivals destined to the particular outgoing link in each time slot (e.g., L may be equivalent to the number of in coming links to the node). We assume that there is no time correlation between cell arrivals in different time slots on each link. In general, it is hard to know a priori the exact probability distribution of cell arrivals. Cell arrivals depend not only on user traffic characteristics but also on jitters introduced by cell multiplexing within the network. If the network provider has no hard data on cell arrival patterns, it can use burst arrival patterns as the worst case (with no time correlation) in terms of buffer overflow, where cells arrive in bursts (L cells) all the time if they arrive. The offered traffic (denoted by ρ) to this link is defined as the average number of cell arrivals in each time slot:

$$\rho = \sum_{i=0}^{L} i\, p_i \qquad \textit{Eq}\ (1)$$

and ρ satisfies the flow – conservation law at each node. At all times, the network maintains ρ<=ρ' by the Virtual Circuit (VC) loss control during cell setup. Looking at the system with a state x defined as i^{th} number of cells in the outgoing link buffer where x=0, 1, 2…B. Such a Markov chain with L=3 and B=7 is shown in fig 3. Let 1/μ and q_i denote the cell transmission time and the steady-state probability of the state being x=i, respectively. 1/ μ is the equivalent to the cell length divided by the link transmission speed, and q_i 's can be easily obtained by solving Markov chains such as that in fig (3). To compute q_i 's, we have to specify probabilities p_i in Eq (3). For instance, for worst case when cells arrive only in bursts of maximal length L, we have $p_L = (\rho/L)$ and $p_o = 1 - (\rho/L)$, so all $p_i = 0$ for 1<=i<=L-1. After computing q_I 's, we obtain u, the average number of lost cells. Using this, we obtain the following [6]:

$$P_l = u/\rho \qquad Eq\ (2)$$

$$D_m = B/\mu \qquad Eq\ (3)$$

Where P_l and D_m denote the cell loss probability due to buffer overflow and maximum cell delay, respectively.

In this paper we analyze the parameters ρ' and B so that we can maximize the maximum allowed link utilization level ρ' while satisfying the constraints on the cell loss probability and maximum cell delay. Note that the parameterρ' is used in the call admission rule. It is clear that, for a given value of ρ, the cell loss probability P_l decreases monotonically as B increases. On the other hand, the maximum cell delay increases monotonically with increasing B. Using these facts, we suggest the following algorithm to engineer B and ρ'.

3.1.1 Algorithm BLS (Backbone Link Sizing)

Step 1: Set ρ':=1
Step 2: Using Eq (3), find the maximum value of B.
Step3:Compute the steady-state probability q of the Markov chain
Step4:Using Eq (2), check for P_l.If the condition in above step is satisfied, terminate the algorithm.
Otherwise, set ρ': = ρ'-0.011 and go to step 3.

Algorithm BLS determines the parameter with a granularity of 0.01 where usually takes a large value. (e.g., =0.96). We can use a finer grid if necessary, at the expense of computation time.

3.2 Markov Chain

The family of random variables forms a stochastic process. The state at time t_k actually represents exhaustive and mutually exclusive outcomes of the systems at that time. The number of states may thus be finite or infinite. For example Poisson distribution represents a stochastic process with an infinite number of states. The important class of stochastic systems includes Markov processes and Markov chains. Markov processes and chains will be useful in dealing with queuing or waiting line theory [7].

4. STEADY STATE ANALYSIS WITH MARKOV CHAIN

For the above example maximum cell delays D_m For different values of cell lengths and buffer sizes were calculated using the relation as below.

Maximum cell delay D_m =Buffer size B/Cell

transmission time μ

Where, Cell transmission time=1/μ= Cell length R/speed

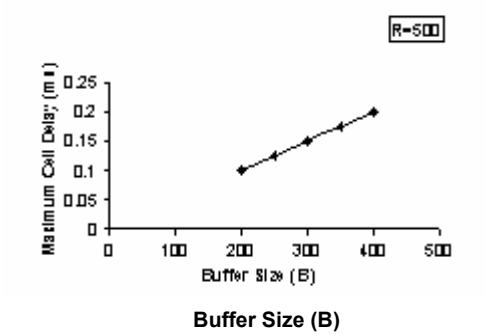
The resultant D_m has been calculated and the values are plotted as shown in fig (1) (The example with R=500 is close to the current ATM standards specification).

For this the above steady value Markov chain model with L=3 and B=7 is assumed. For these, steady state probability values p_0, p_1, p_2, p_3, were computed for uniform and bursty traffic. The values are plotted as shown in fig (2). As is expected, bursts arrivals experience higher cell loss probability for the maximum link utilization level (ρ_l) and buffer size (B) These are independent of the maximum link utilization level. Note that, for this range of buffer sizes, the maximum cell delay is less than 1 ms due to the high transmission speed of links. We use algorithm BLS to size the parameter values for different constraints for uniform and bursty arrival patterns, respectively. In this example, the cell loss probability constraint of 10^{-11} means that (on the average) one cell is lost due to buffer overflow out of 10^{11} cell arrivals. Maximum cell delay value is calculated for various buffer sizes. Then the cell transmission time is varied and maximum cell delay is calculated. For uniform cell arrival the Markov chain model for B=7 and L=3 with the probabilities p0, p1, p2, p3 has been computed (p0=p1=p2=p3).For bursty cell arrival the probabilities p0, p1, p2, p3 has been computed (p1=p2=0).The packet loss probability is calculated using Eq (2) for both uniform and bursty traffic. The value of traffic load is changed and corresponding cell loss probability is computed.

$$\%BECN\ Cells = [N/(768 + T_d)] \times 100 \qquad Eq\ (4)$$

Where N - Number of Sources.

T_d -Transmission Delay.



Buffer Size (B)

Fig 1 Maximum Cell Delay

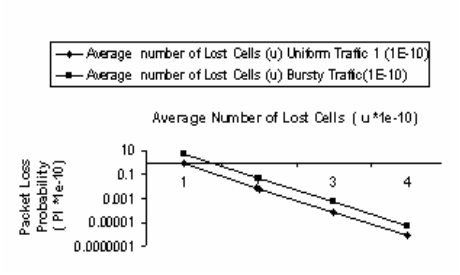


Fig.2 Comparison of uniform and busbursty traffic

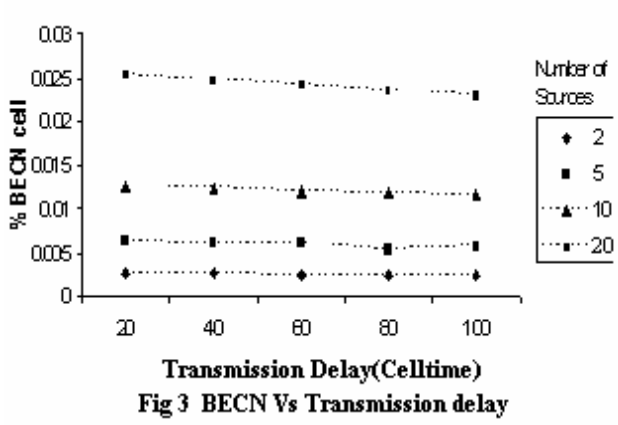


Fig 3 BECN Vs Transmission delay

5. BECN SCHEME

5.1. Feed Back Mechanism

Feedback schemes use a closed loop feedback control mechanism that allows the network control the cell emission process at each source. Each virtual connection must have an independent control loop since each connection may have different path through the network. Two classes of feed back scheme have been proposed namely credit based and rate based. The Credit-based scheme is a link-by-link window flow control scheme. Each link in the network independently runs the flow control

The Rate based schemes use feed back information from the network control the rate at which each source emits cells into the network on every virtual connection. Three types of rate-based scheme have been proposed:

Explicit Rate Control, Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion

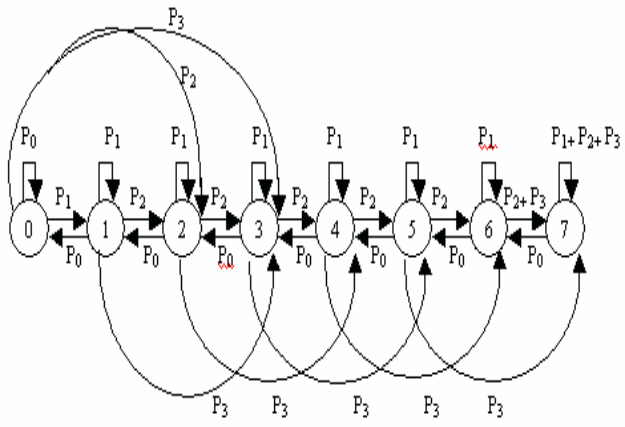


Fig 4 Markov Chain When *L*=3 and *B*=7

Notification (BECN). With explicit control the network periodically determines at what rate each source should be transmitting and sends a message to each source informing them of the new rate.

5.2. Fecn Scheme

FECN is an end-to-end scheme in which most of the control complexity resides in the end systems. When a path through a switch becomes congested the switch marks a bit in the header

Increasing factor

Increasing Factor = (Min., Speed / Max., Speed) *

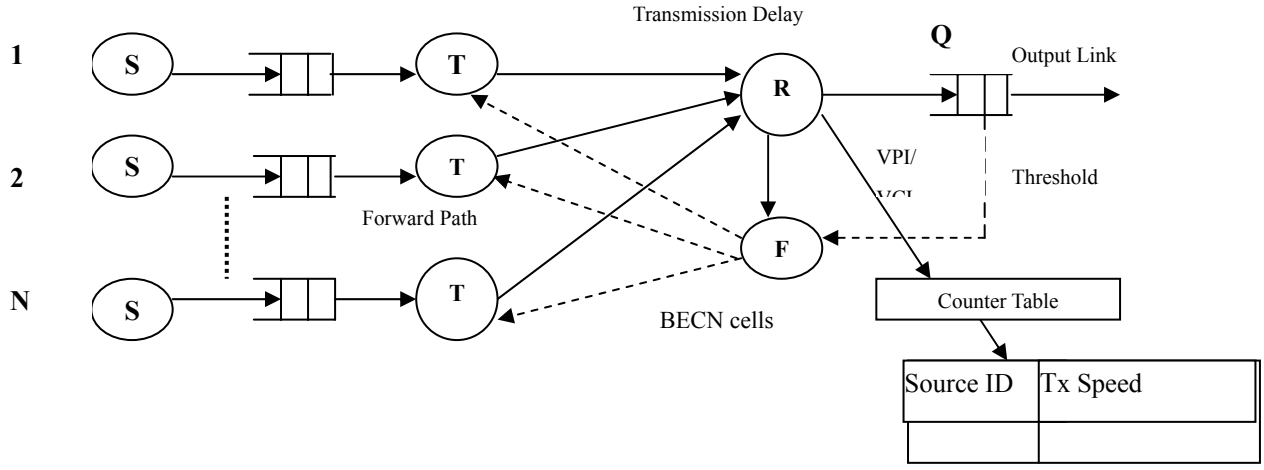


Fig 5 Modified BECN Scheme

Congestion Status Table

of all cells at that path in the forward direction to indicate congestion. The destination end system monitors the congestion status of each active source and sends congestion notification in the reverse direction on each active virtual connection to inform the source of the congestion status. The source uses this feedback to increase or decrease the cell transmission rate on each virtual connection.

In BECN congestion information is returned directly from the point of congestion back to the source for each virtual channel. The source sends the cell on each virtual connection in a similar manner to FECN. In the following discussion we present a BECN mechanism for ATM LAN with some simulation results for a simulation model and further results are analyzed.

5.3 Modified Becn Simulation Model – Kvs Scheme

The simulation model of the BECN scheme is shown in fig 6. The source S generates packets that are queued for transmission by the transmitter T. The transmitter represents the segmentation process for a specific virtual connection in the host's ATM network interface, which segments packets into cells for transmission over the ATM network. R is a point of congestion in the network

Here a Congestion Status table is maintained at Receiver. This table consists of Speed of Transmission and corresponding Source ID of corresponding Transmitting Source.

If the length of the queue Q exceeds the threshold, the filter F will send congestion notification (BECN) cells back on the virtual channels. At this juncture the speed of transmitter is reduced in a dynamic approach. This approach avoids the fixed increasing and decreasing approach as per peter Newman model[8]. We proposed the new KVS algorithm in which we compare the maximum speed to minimum speed and reduce the speed of Transmission.Here the increase or decrease in speed is not on fixed term and this term is varied as per the min/max factor.

Decreasing Factor= (Max., Speed / Min., speed) *

speed of transmission

If the congestion is not cleared the above procedure is repeated continuously. If the congestion is cleared the speed of transmission is increased dynamically by calculating the

Speed of transmission

The congestion is checked and above process is repeated. Since we proposed it, named as **KVS**

(**K**arthikeyan, **V**asanthan, **S**anthana Krishnan)

KVS Algorithm

Step 1: The Source and speed of transmission is being maintained in the table.

Step 2: The length of Queue is fixed.

Step 3: Check whether any congestion notification from the filter, if Yes go to step 4 else Continue the transmission and step 3 is repeated.

Step 4: Calculate the maximum speed and minimum speed of Transmitter in the table.

Step 5: The speed of transmitter is reduced dynamically calculating the Decreasing factor

*Decreasing Factor= (Max., speed / Min., speed)**

speed of transmission

Step 6: Check whether congestion is cleared, if not do step 4 else speed of transmitter is increased dynamically by Increasing Factor

IncreasingFactor= (Min., speed/Max., speed)*

speed of transmission

Step 7: Repeat the Step 6 process.

5.4 Performance Results

(i). The maximum cell delays (D_m) for different values of cell lengths R for various buffer size have been calculated. The result shows that cell delay increases with increase in buffer size and cell length as in fig (1).

(ii). Using Markov chain cell loss probabilities for uniform and bursty cell arrival pattern were calculated. The bursty cell arrivals experience higher cell loss probability than uniform cell arrival pattern as in fig (2). It supports the classification of two different traffic namely Guaranteed traffic and ABR traffic by ATM Forum.

(iii). The Analytical model of BECN scheme results is shown in fig (4).

(iv). The simulation model of BECN scheme of Peter Newman approach using C coding produces a result, which

are given in fig (7).

(v). The comparative analysis has been made and the simulation model for the percentage of BECN cells are almost close to the analytical model upto 10 sources.

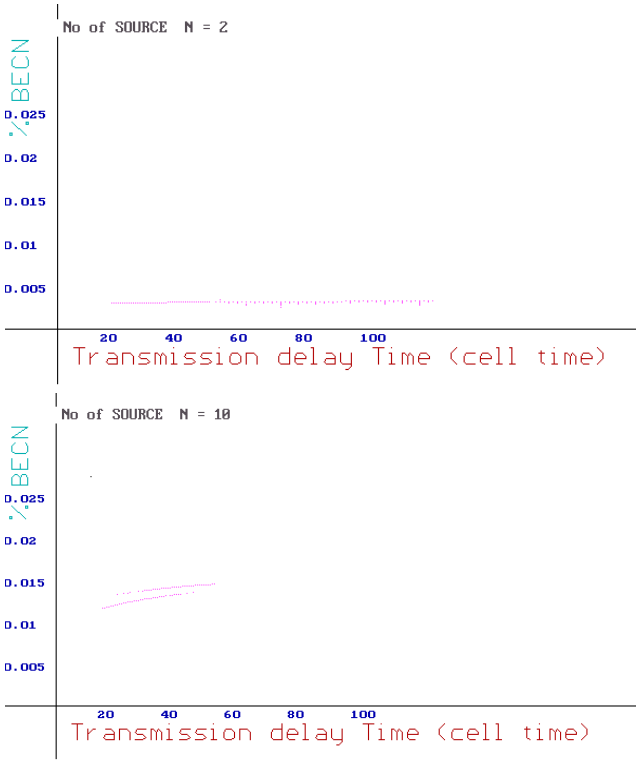


Fig 7 Simulation Results

6. SCOPE FOR FUTURE WORK

The proposed KVS algorithm's simulation is to be implemented. The simulation model may be implemented as a switching architecture using VLSI technology. This would minimize the various delays involved in ATM networks. Simulation of VLSI will also validate proper hardware implementation. Also Genetic Algorithm (GA) techniques may be incorporated to improve the performance analysis of BECN scheme.

7. CONCLUSION

The performance of BECN with the analytical model suggests that for non real-time bursty traffic such as data cannot be accommodated in the existing service category with the implied QoS. From this it is inferred that a separate service namely ABR will be better suitable for bursty traffic.

From the figure (4) and (7) it is found that the performance of analytical model and simulation model are very similar. When number of sources exceeds 10 then the

BECN scheme can be employed for non real-time bursty data services. It has been implemented using C.The KVS algorithm model namely Modified BECN Scheme is to be implemented and its simulation model has to be compared with BECN Scheme.

8. ACKNOWLEDGEMENT

We extend our sincere thanks to KVITT Management, Principal, Head of the Departments of Information Technology, Electronics and Communication and Asst.Prof.N.K.KarthiKeyan, Department of ECE for their encouragement and support in our endeavor.

9. REFERENCES

[1] S. Keshav, An Engineering Approach to Computer Networks, Addison Wesley, 2000.

[2] S.P.Singh et al , An Overview of Congestion Control Techniques in ATM networks and some performance results IETE Technical Review,Vol 17,No. 3, pp 87-103, May-June 2000.

[3] William Stallings, Data and Computer Communication, Pearson-Asia Edition, 2001.

[4] R.Jain, Congestion Control in Computer Network: Issues and Trends, IEEE Network, pp 24-30, May 1990.

[5] H.T.Kung and Robert Morris, Credit based Flow Control for ATM networks, IEEE Network, pp 40-48, March/April 1995.

[6] A.Gersht & Kyoo.J.Lee, A Congestion Control Frame work for ATM networks, IEEE Journal on Selected areas in Communication, Vol 9,No 7 pp 1119-1129 September 1991.

[7] Hamdy A.Taha, Operation Research, Macmillan Publishers, 1989.

[8] Peter Newman, Traffic management for ATM Local Area Networks,IEEE Communication,Vol 32,No 8,PP 44 -50,Aug 1994

Single-Copy UDP Protocol for Cluster System

Lan Gao, Hai Jin and Zongfen Han
Huazhong University of Science and Technology
Wuhan, 430074, China
E-mail: hjin@hust.edu.cn

ABSTRACT

With the development of high-speed network technology and the advances of microprocessors, cluster system is becoming the mainstream technique of parallel computing. In workstation cluster system, there is much lower error rate, which makes UDP an alternative communication protocol for effective scalable parallel computing system.
Multiple factors that lead to the inefficiency of cluster communication are presented and analyzed prior to design and implement a general kernel-level light-weight communication mechanism based on single-copy UDP protocol. The mechanism supports nearly all commercial components and reduces the communication software overheads by eliminating data copying from user space to kernel space, simplifying the protocol processing flow and exploiting a new buffer management method.
Three efficient methods are presented to provide a faster data path between an application and the network: the kernel-level protocol, the memory mapping mechanism, and the buffer management mechanism.
A prototype system is implemented and the performance analysis is provided. The results prove that single-copy UDP protocol over Fast Ethernet can exploit the performance of modern networks by using techniques mentioned above. It not only provides low latency and high bandwidth communication, but also guarantees the stability of the communication.

Keywords: Cluster System, Communication Protocol, Communication Overhead, Single-Copy, User Datagram Protocol

1. INTRODUCTION

The TCP/IP protocol stack has gained remarkable achievement during the dramatic expansion of the global Internet. However, with the emergence of the high-speed transmission facilities and the novel-style application requirements in cluster system, TCP/IP also shows some built-in drawbacks in its functionality and performance. It is too complicated with the error-recovery mechanism, buffer management mechanism, timer management mechanism, and flow control mechanism. With much lower processing overheads, UDP beats TCP in the cluster environment and we focus on UDP communication.
In cluster system, the network environment has changed from Internet with low bandwidth and high error-rate to Fast Ethernet or Gigabit Ethernet with high bandwidth and low error-rate, and applications rely more on performance rather than reliability. Thus, UDP is a reasonable alternative for cluster communication instead of the complicated TCP protocol. Although UDP provides unreliable service, it fits well with the characteristics of multimedia communication in cluster video servers.
There exists large amount of data transmission in cluster system such as cluster video server, for example, a one-minute movie segment probably occupies 18MB disk space and requires 2 seconds to transfer between central control node and one of the storage nodes. It is necessary to improve the communication bandwidth so that more segments could be distributed to different storage nodes at the same time. These characteristics demonstrate that it is possible to construct a simpler and more efficient UDP transmission mechanism by cutting the data-copying overhead.
The remainder of this paper is organized as follows. Section 2 explains the background and related work. Section 3 gives an analysis of the overhead of normal UDP communication. Single-copy technique is introduced in Section 4 in order to reduce the per-byte overhead. The implementation method of Single-copy UDP protocol together with some core algorithms are also discussed. The results of performance experiments of Ping-Pong tests are shown in Section 5. The paper concludes in Section 6.

2. RELATED WORKS

There have been several attempts to reduce the number of data copying in networking research field [1]. Because the communication software overhead is several orders of magnitude larger than the hardware overhead, modern improved protocols are designed to avoid the time-consuming communication path from application to kernel and to network device [2].
The most popular approach is zero-copy technique. By exploiting advanced hardware capabilities (e.g. network devices with enhanced DMA engines and co-processors), much of the functionality originally executed by operating system is moved into the hardware device. Thus, applications can interact directly with the network interface avoiding any system calls or kernel interaction. In this way, the processing overhead is considerable reduced providing both reduced latency and higher throughput. There are several approaches to providing low-latency and high bandwidth communication, for example:
Fast Messages [3] is a high-speed messaging layer that delivers low latency and high bandwidth for short messages over a Myrinet on a NOW. FM simplifies the communication critical path by moving as much functionality as possible into the application layer and the network co-processor. Fast Sockets exports the Berkeley Sockets programming interface using a high-performance protocol, which collapses and simplifies protocol layers and transfers some of the protocol knowledge required into user-level programming.
However, such an approach relies much on the expensive and special hardware because it needs the virtual memory protection mechanisms to maintain system security and integrity. Hardware solution means an additional cost which damages the flexibility of the cluster system. Furthermore, the hardware network interface may not be used with standard I/O buses.

3. OVERHEAD ANALYSIS OF TRADITIONAL UDP PROTOCOLS

The overhead of UDP is divided into two types: per-packet overhead and per-byte overhead [4]. Per-packet overhead is associated with the handling of each packet, and is proportional to connection times. Per-byte overhead is accumulated with every byte in a packet, and is proportional to the size of packets.
The inner control flow of UDP transmission implemented on the Linux operating system is explained in Figure 1. The UDP send primitive is a system call associated with sockets. When send is invoked, the control is transferred to BSD socket layer with appropriate parameters. Then, the socket layer calls the address family interface, in most cases the Internet Layer. Internet layer will call the appropriate protocol layer, which in our case will be UDP and IP layers. The protocol layer will build the protocol headers and make packets. It forwards the packets to the generic device driver that processes the packets device independently. Finally the generic device drive passes the packets to the network interface driver.

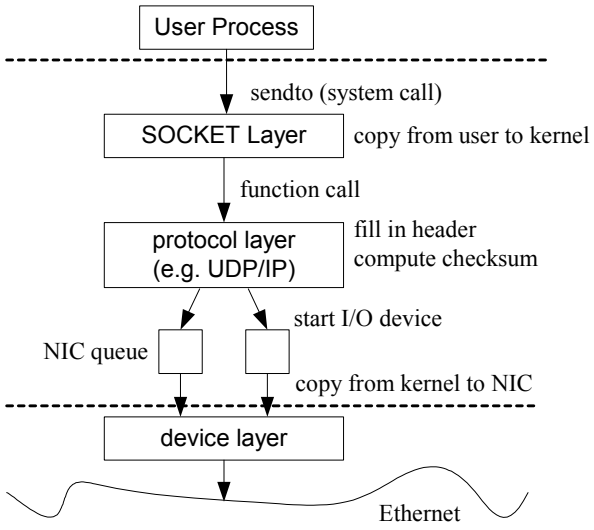


Figure 1 UDP Control Flow

Generically speaking, the overheads of the UDP protocol consist:

- context switching overhead between the system space and the user space when the applications call the system call SEND to transmit data
- data copying overhead between the user space and the system space, and the overhead of data copying between two adjacent protocol layers, e.g. when the data needs to be segmented
- overhead of analyzing data coming from adjacent protocol layer, the overhead of data partitioning and reconstructing, e.g. in IP layer
- overhead of fill in packet heads added by different protocol layers, and the overhead of computing checksum
- overhead of routing, connection maintaining, traffic controlling, error detecting and recovering and buffer management
- overhead of starting I/O device, e.g. NIC, and the overhead of data copying between the system space and the device buffer

The per-packet overheads include operating system overheads and protocol processing overheads. The per-byte overheads include data copying overheads and checksum computing overheads. To be more concrete, operating system overheads include time spent in system call, context switching, interrupt processing, swapping, and starting I/O device, etc. Protocol overheads include buffer management, connection maintaining, error controlling, flow controlling, routing, etc. Data copying happens mainly in three places: from user space to kernel space, from kernel space to network buffer, and probably copying in kernel space when data needs to be partitioned and reconstructed.
The goal of this paper is to reduce the number of data copying, thus reduce the per-byte overheads which take the most part of the overheads in data transmission of large amount of data like multimedia data.

4. SINGLE-COPY UDP PROTOCOL APPROACH

As mentioned in Section 3, reducing the number of data copying is a key to minimize the per-byte overhead. This paper introduces a new approach for the UDP protocol that eliminates the data copying between user buffer and kernel buffer. Figure 2 explains the new semantic.

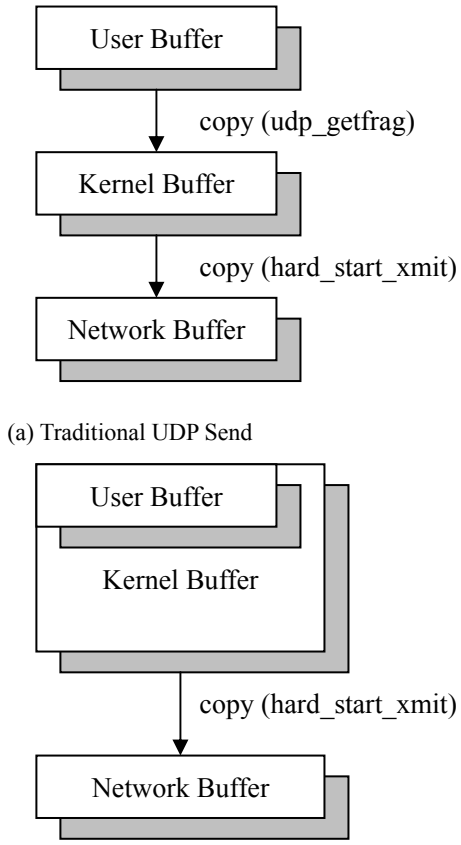


Figure 2 Comparison of Traditional UPD Send and Single Copy UDP Send

Traditionally, there are two data copies after sendto is called. One is between user buffer and kernel buffer; the other is between kernel buffer and network buffer. User buffer is allocated statically by declaring array of certain size, or dynamically by calling malloc function. User processes use virtual address to visit this area, however, continuous virtual memory does not mean continuous physical memory, so consecutive virtual addresses do not correspond to consecutive physical addresses. Addresses are translated by querying the page tables.

Single-copy approach maintains the traditional control flow, but eliminates the data copying between user buffer and kernel buffer by "merging" two buffers together, as Figure 2(b) shows. Design issues are all related with the shared memory area, which consist of the creation and destruction of the shared memory, the address mapping mechanism and the buffer management mechanism. The structure of single-copy approach is showed in Figure 3

SHM is used to refer to the shared memory area. SHM is closely related to kernel module technology. Firstly, the char device kmmapdrv is installed and registered with the operation set (open, release and mmap) in the module, and is unregistered when uninstalling the module. Next when initializing the device, SHM is created as the communication area between user and kernel. It is refreshed each time when user process requires address mapping, and is freed when uninstalling the module.

When user process is to send data, it creates a special device file and uses the device operation mmap to get the user buffer address. Actually, mmap finds the free user address and construct page tables to create mapping between this address and the kernel address of SHM. Then the user address is returned to user process as the send buffer address. Because SHM is shared by user and kernel, data written by user can be directly accessed by kernel. No copy is needed between user and kernel.

4.1. Memory Mapping Mechanism

Memory mapping mechanism is the basis of the single-copy sub-system. Its main goal is to allocate physical memory with continuous physical addresses by kernel through kmalloc function. By using mmap method from the char device operation set, the area is mapped to the virtual address space of each user process, which leads to the sharing of the same memory area between user and kernel. The creator is kernel, and the address translation requester is user. The merit of such design is that the SHM area is fixed in the kernel space and user won't lose the mappings even process switching happens. Another merit is that user can use SHM as easily as traditional way, it needs not link special costumed function library, only to replace malloc with mmap.

The linear address space is separated to several areas, as Figure 4 shows. Area from 0GB to 3GB is called user space, and it is local to each user process, which means that the same user virtual address may be mapped to different physical address. Area from 3GB to 4GB is called kernel space, and it is global, which means that all user processes can access it with the same linear address. Kernel space is divided into two areas, K Area and V Area. Each address in K Area is mapped to one single physical address with the difference equal to PAGE_OFFSET. If kmalloc function is called, a K Area address is returned. On the contrary, physical addresses corresponding to V Area addresses can only be got by inquiring kernel page tables. V Area address is returned by vmalloc function.

We use kmalloc to allocate a continuous area in K Area of kernel space. Because addresses in K Area are fixed in the 4GB linear address space, different user processes can access this area with the same address even if there happens context switching or other change of process environment. By implementing a special mmap method of char device kmmapdrv, new page table items are constructed which create new mappings between kernel addresses of K Area and user addresses of Set A. Thus, user process can use its own valid linear address to visit the kernel space area, which makes the communication between user and kernel possible.

When mmap is called by certain user process, the operating system will search the virtual address space of current process for unmapped addresses. These addresses are fed into the mmap implementation as the map source, and the addresses returned by kmalloc as the map destination. Mmap implementation creates new page directory entries and page table entries to create mapping relation between them and finally valid user addresses are returned for use in user space.

Traditionally, data are copied from user space to kernel space in the function udp_getfrag() during the sendto process, but by using memory mapping technique, the same physical address can be accessed by one kernel linear address and one user linear address, there is no need to copy data again, as Figure 5 shows. This is how single-copy mechanism is implemented.

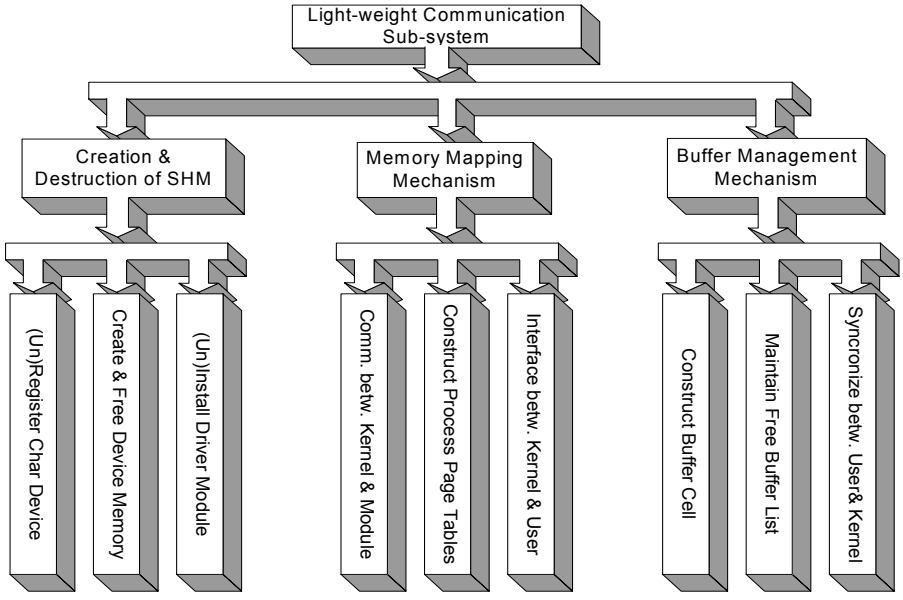


Figure 3 Structure of Single-Copy Approach

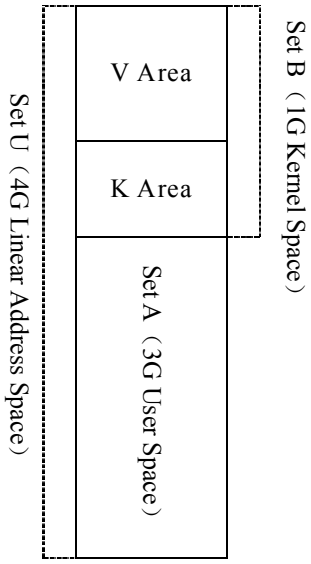


Figure 4 Linear Address Space

The SHM is a large continuous area, while each sendto needs only a fraction of it. If use the whole area as one large buffer, data transmission speed is restricted and may lead to low efficiency; if the area is divided into several small cells with different length, new overhead will be introduced to the communication among user, kernel and module. We introduce a buffer management mechanism, which separate the SHM into cells with equal length for faster communication and easier management.

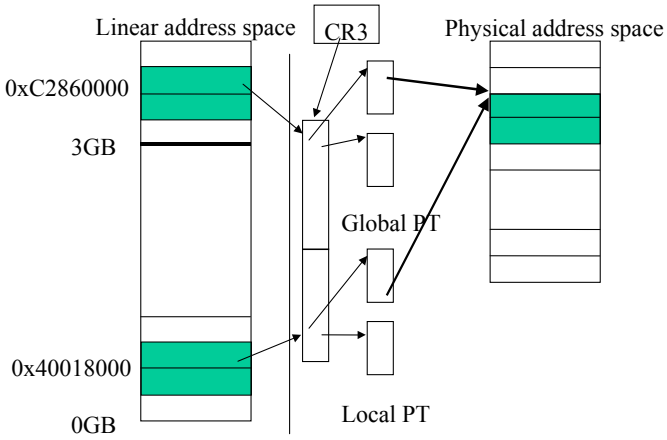


Figure 5 Memory Mapping Mechanism

4.2. Buffer Management Mechanism

The inner structure of SHM is showed in Figure 6. Hbuf contains two pointers: head and tail. Head always points to the first free cell, while tail points to the last free cell. Each cell consists of the following parts: Hcell which points to next free cell, Hhh which contains the Hard Header (e.g. Ethernet Header), Hip which contains IP header, Hudp which contains the UDP header, and the real data. Considering the CPU cache length and the maximum UDP packet length, we define cell length as 2K, thus total buffer length is 8+n*2K.

Hcell of each data cell forms a virtual queue which links all free data cells, as figure 7 explains. Each time when user calls the mmap function, it gets the head pointer, and uses content of head pointer as send buffer address. The organization of virtual list is transparent to user process. Except for the mmap system call instead of the traditional malloc, user program need not other changes. Kernel and module have total control of the virtual list. The separation of head and tail pointer synchronizes different kernel parts accessing the virtual list at the same time, e.g. the detach and the attach process of free cells. Each time a cell is detached from the virtual list, head pointer is modified to points to the content of the current cell's Hcell area. Next when this cell is to be attached to the virtual list, the cell pointed by tail points to this new cell, and then tail pointer is modified to point to the new cell.

Actually, the content of each Hcell and Hbuf contains only the offset to the first byte of SHM. This is because user and kernel uses their own linear address to visit SHM, only the offset is the same value.

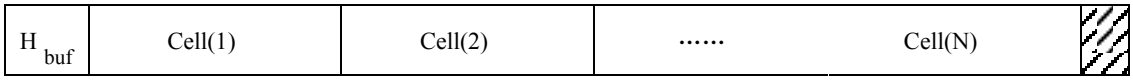
4.3. Implementation

The system is implemented in three levels: user level, module level and kernel level, as figure 8 shows. The module is responsible for allocating memory as SHM, formatting this area as figure 6 and notify kernel to use it as the network send buffer. When user is trapped into kernel, module finishes the memory mapping process for it and sets the map flag to notify kernel to replace original non-single-copy method. User process uses mmap system call to get the send buffer address, and then transmit data by calling sendto. The kernel updates the virtual free cell list and decide when and how to send the real data. In this process, data copying from user to kernel is avoided.

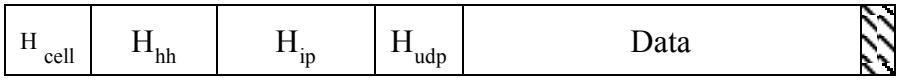
5. PERFORMANCE ANALYSIS

Our implementation and experimentation environment consists of two PCI PCs connected by a Cross-Over Cable via 3Com Network Interface Cards (3Com 3C905B), as Table 1 shows.

We use standard Ping-Pong test to evaluate the UDP and single-copy UDP network latency and the bandwidth. Latency is defined as the time elapsed when messages travel from the sender side to the receiver side. Bandwidth is the bytes transmitted per second. Both sender and receiver are user-level processes.



(a) Buffer Organization （Size = L_{buf} ）



(b) Cell Organization （Size = L_{cell} ）

Figure 6 Buffer Structure

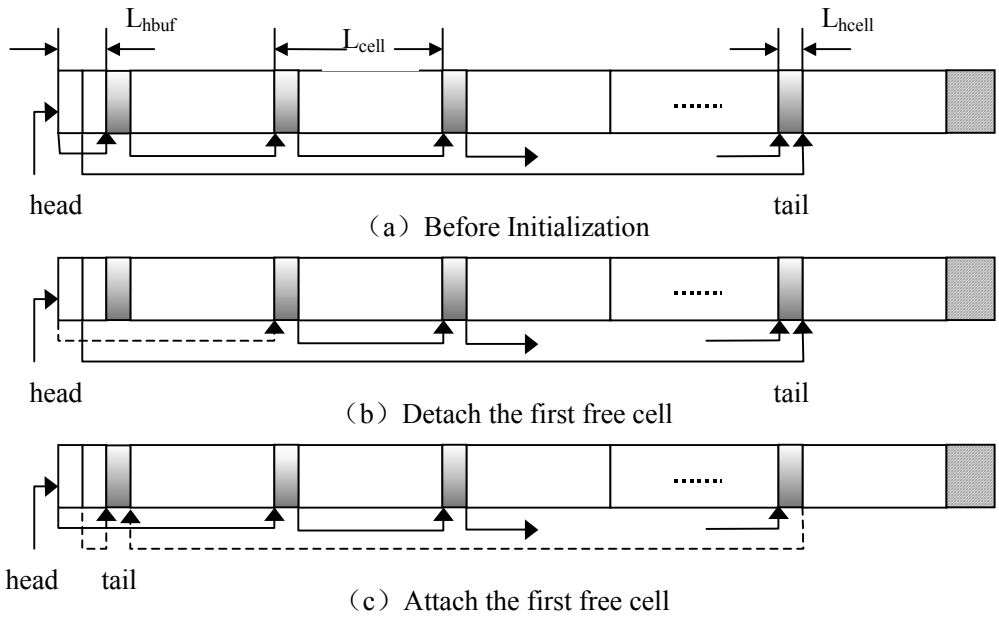


Figure 7 Free Cell List

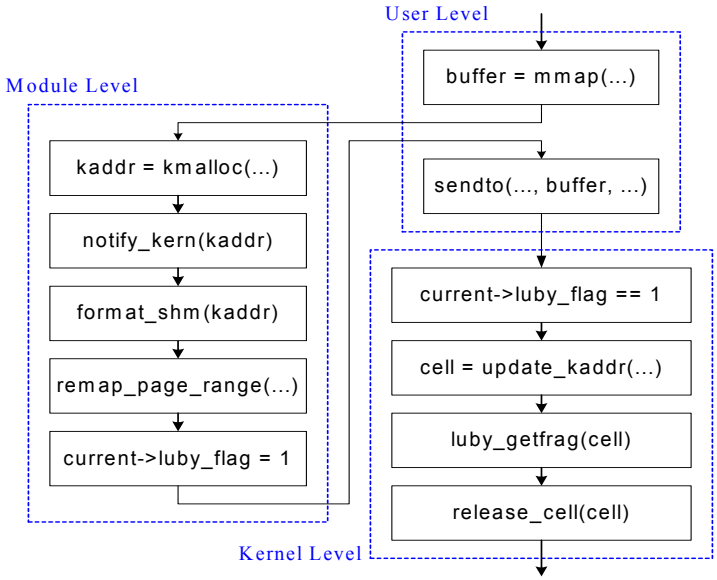


Figure 8 System Implementation

The master node sends a ping message to the slave node first. After receiving this ping message, the slave node sends a pong message back. Actually, we get the round-trip time through several ping-pong tests, and the latency is computed as half of the average value of the round-trip time. Results are showed in Figure 9 and Figure 10, respectively. Because of the simplification in data copying and buffer management, single-copy UDP has much lower latency than traditional UDP protocol, especially when sending large amount of messages. For example, when sending messages of 64 bytes, latency is 83μs vs. 87μs, and improvement is only 1%. While sending messages of 1280 bytes, latency is 401μs vs. 422μs with the improvement equal to 5%.

In the prototype implementation, no modification is applied to the data fragment and defragment, thus the maximum message length is only 2KB. In order to send large messages, we send several fix-length short messages to emulate large data transmission. Figure 10 shows the transmission rate in our test.

6. CONCLUSIONS AND FUTURE WORKS

This paper introduces a single-copy UDP approach to improve the performance of large data communication. It allocates a kernel buffer and maps this area to user space, thus eliminates one data copying in the UDP send. The approach has been implemented in the Linux operating system. The results of experiments show that latency improves over 5% higher than the traditional UDP send for packets of 1280 bytes or more. Considering that the data size in cluster video server, communication is typically several kilobytes, our experiment results demonstrate that the single-copy UDP send can be very effective for multimedia communication.

In our first-stage implementation, performance improvement

is largely based on the latency reduced in data copying from user buffer to kernel buffer. Actually, there may still exist data copying in kernel because of the data fragment and defragement. Our next goal is to modify the buffer management mechanism to make possible the transmission of data larger than 2KB.

Table 1 Testing Environment

	CPU	Memory	NIC	PCI Bus	OS	Connection
Master	Intel PIII 500MHz 512K Cache	128MB PC100 SDRAM	3Com 3C905B 100BaseTX	32-bit 33MHz	RH Linux 7.1 Kernel Version 2.4.2-2	Cross-Over Cable
Slave	Intel PIV 1.4GHz	256MB PC133				

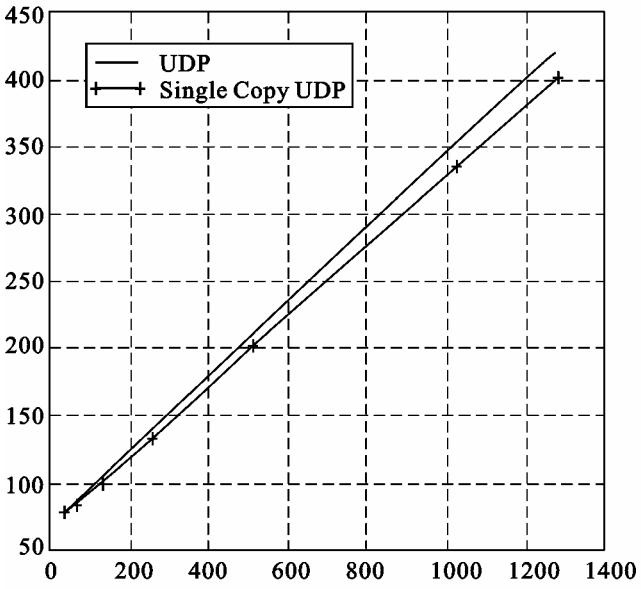


Figure 9 Latency

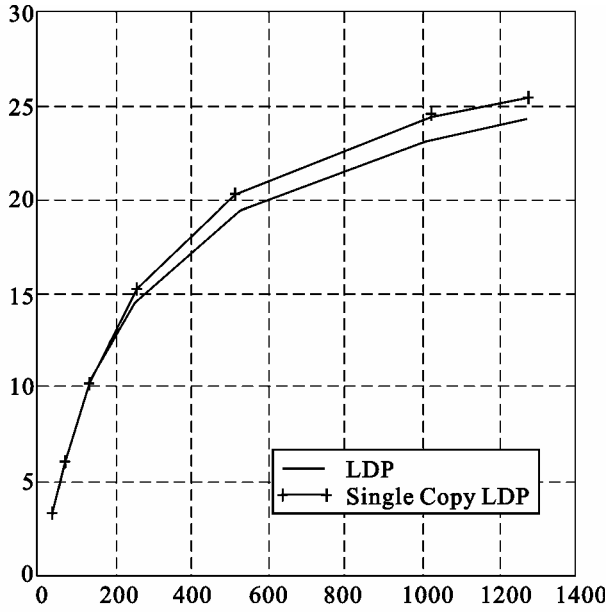


Figure 10 Bandwidth

7. REFERENCES

[1] R. Buyya, High Performance Cluster Computing: Architectures and Systems, Vol.1, Prentice-Hall, Inc., 1999, pp.182-184.

[2] P. Melas and E. J. Zaluska, “Performance of Message-Passing Systems Using a Zero-Copy Communication Protocol”, Proceedings of International Conference on Parallel Architectures and Compilation Techniques, 1998, pp.264-271.

[3] S. Pakin, V. Karamcheti, and A. Chien, “Fast Messages: Efficient, Portable Communication for Workstation Clusters and MPPs”, IEEE Concurrency, April-June 1997, Vol. 5, No.2, pp.60-72.

[4] D. Clark, V. Jacobson, J. Romkey et al., “An Analysis of TCP Processing Overhead”, IEEE Communications Magazine, June 1989, Vol.27, No.6, pp.23-29.

Airline Ticket Reservation System Using Mobile Agent

MIN, YU [1,2] and JINYUAN, YOU[1]
1. Dept. of Computer Science & Engineering, Shanghai JiaoTong University, Shanghai, 200030, P.R.China
E-mail: myu821@163.com

DINGKANG, ZHOU[2]
2. Institute of Computer Science&Technology, Jiangxi Normal University, Nanchang, 330027, P.R.China

ABSTRACT

This paper describes a distributed system simulating a simple airline ticket reservation system. We introduce the mobile agent techniques to distributed database, and a principle of "code moving to data" to achieve efficient communication through the network. It also implemented the inter-agents communication.

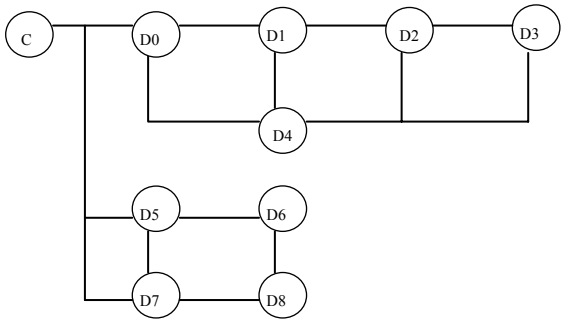
Keyword Mobile agent, Distributed database, Simulation

1. INTRODUCTION

With the emerging need for real-time online transaction systems for buying and selling on the Internet, some are creating a mobile agent-based electronic market place, mobile agents offer several good reasons for use in this arena. Mobile agents are entities that navigate autonomously through the network based on their own internal program and state. The agents coordinate the computation in both time and space to solve one large problem, it is a future distributed computing model[1,2,3]. The mobility of agent is more suitable for the characteristics of the distributed application, it reduces the complexity of programming and semantic gap between abstract algorithm descriptions and their actual implementations. In this paper, we introduce mobile agent technique to distributed database, create a distributed system simulating the airline ticket reservation system, which is implemented by mobile agent. It supports cooperative work among agents. The system is intended to help a ticket agency in seeking the best deals given the client's specific requirements and constraints.

2. SYSTEM DESIGN

Airline ticket reservation system is inherently distributed. As a ticket agency, which sell all kinds of domestic or international tickets, it can be the agency of many airlines. The information about the tickets of all these airlines should be gathered to provide the customer a best selection, i.e. with lowest price and fastest time for the travel. However, each airline has its own price, its own schedule, and its own database to manage its tickets. The status of the tickets include departure, destination, time, price and availability, change constantly. So it is sensible to keep track of the information at each airline on its own computers rather than put them on a central machine. As a result, the task is to book the cheapest ticket to satisfy the customer's requirement. The most important feature of distinguishing agent-based systems from other conventional systems is that all functionality of the application is embedded in individual agents, i.e. the programs are carried by agents as they navigate through space. By using agents, the airline ticket reservation system is more conveniently simulated by nodes and more easily implemented by agents.



Note: C represents the Console node
D* represents the Database nodes
Figure 1 The graph of nodes

We simulate the structure as the graph of nodes (Figure.1), the actual situation may differ very much. The ticket agency server is represented by the Console node named "C". The airline database servers are represented by Database nodes named "D*", where * represent the number of the node. The functionalities of the system are implemented by client_agent and cyber_agent separately. The functions of the client_agent are to get customer's requirement and test its correctness, send requirement to the cyber_agent, accept the result of the suitable ticket from the cyber_agent, list the lowest price tickets to the customer, send the customer's selection to the cyber_agent to book the ticket. The functions of the cyber_agent are to accept the customer's requirement from the client_agent, move to all the Database node to find suitable tickets for customer in the database, send the result of the tickets to the client_agent, accept customer's selection from the client_agent, move to the specific Databas node to book the required ticket for customer in database. The client_agent and the cyber_agent communicate on the Console_node, the cyber_agent get requirement from the client_agent and return the result to the client_agent on this node by using Rendezvous. The client_agent also interact with the customer on this node, such as the requirement, list of tickets, booking selections are all shown here.

This graph of nodes can be extended to any shape of graph consisting of any nodes and links. There are many operations concerned with the customer. The interaction with the customer are not the advantage of agents, since these functions are not distributed. There are also many operations are distributed and need the agents, so the next section will describe the functions related to agents.

3. IMPLEMENTATION

The system is just to simulate the airline ticket reservation system, the functions for customer are not full. In addition, the database is displaced by text files for convenience in the implementation, and the records in the files are limited. This system implemented in a virtual infrastructure for running mobile agent, called MESSENGERS[2].

Inter-agents communication

The client_agent and cyber_agent communiate on Console node. First, two agents can change data by using node variables. Second, the client_agent and cyber_agent can be synchronized by using Rendezvous-style technique, which makes a output action and the corresponding input action occurring simultaneously.

(1)When client_agent pass the requirement of the customer to the cyber_agent:

client_agent-- requestRendevous(hand_in_requirement, &require)

cyber_agent-- acceptRendevous(hand_in_requirement)

Note: at first only one cyber_agent waiting for the requirement of the customer.

(2) When cyber_agent pass the result of tickets to the client_agent:

client_agent -- acceptRendevous(for_ticket)

cyber_agent--requestRendevous(for_ticket, &ticket_pack)

Note: since at the same time there are several cyber_agents to find the tickets for the customer, the client_agent must accept all the results of them. So the customer call acceptRendevous() function as many as the cyber_agent numbers and each cyber_agent call requestRendevous() function one time.

(3) When client_agent pass the selection of tickets to the cyber_agent:

client_agent -- requestRendevous(hand_in, &number, &ticket_pack)

cyber_agent -- acceptRendevous(hand_in)

Note: this time only one cyber_agent to book the ticket for the customer.

Traversal of the database nodes

When the cyber_agent find the suitable tickets for the customer, The system uses Depth First Traversal to implement the traversal of the graph of the database nodes(Figure.2). The condition is that all nodes know their neighbors. The implementation of information gathering is in Figure.3.

```
while ( !go back to the console node )
    { if ( node is visited )
      {  //may be visited by other cyber_agent
         go back to the node where it comes;
         }
     else
      {  node is visited;
         find the ticket in the database on this node;
         node← the node's first unvisited neighbor;
         if ( an unvisited neighbor node is found )
            {go to the neighbor node;}
         else { //all the neighbors are visited and node is
NULL
               go back to the node where it comes;}
               }
    }
```

Figure 2 Traversal of the graph of the database nodes

```
      nodeX = the console node;
      while ( node is not visited )
       {
         record the path come from nodeX is the best path
of current node;
         nodeX = the current node;
         hop() to all neighbors of the current node--nodeX;
         // here, when hop to all neighbor nodes,
         // agents are made copies on each of the neighbor
nodes
         node← neighbors of nodeX;
         on all these new nodes record they has a neighbor
nodeX;
      }
```

Figure 3 The neighbor's information gathering and the path

Path to a specific node

When the cyber_agent book the ticket, it should go to the node which contains the ticket directly. The system think when a node is first time visited, the path from the console node to it is the best one (Figure.3).

4. RESULT

The system should be run by four step: (1) generate the graph of the nodes; (2) initialize some information like neighbors of a node and path to the Console node; (3)inject the cyber_agent on Console node; (4) inject the client_agent on Console node. The cyber_agent use a rotation to continue serving the customer, so that the client_agent may be run as many times as you like.

The Console information that client_agent is interacting with customer:

Please input the DATE of your travel in format of mm-dd-yyyy:

01-01-2003

Please input the name of your DEPARTURE place:

Beijing

Please input the name of your DESTINATION place:

Shanghai

Please input the number of tickets you want to list:

3

User Requirement:

Date: 01-01-2003

Departure: BEIJING

Destination: SHANGHAI

Count: 3

Suitable Tickets found:

No. 1:

Ticket ID: pe0012k3324034

Date: 01-01-2003

Leave at 6:00 from BEIJING

Arrive at 8:00 to SHANGHAI

Price is $750.00

Left 7 tickets

No. 2:

Ticket ID: pa0012k3324034

Date: 01-01-2003

Leave at 6:00 from BEIJING

Arrive at 8:00 to SHANGHAI

Price is $750.00

Left 7 tickets

No. 3:
Ticket ID: pb0012k3324034
Date: 01-01-2003
Leave at 6:00 from BEIJING
Arrive at 8:00 to SHANGHAI
Price is $750.00
Left 5 tickets
Then you can book the ticket.

Please input your choice of ticket(1~3):
2
You have selected No. 2.
ok, ticket booked.

Using mobile agent-based approach to simulate the E-commerce is totally different from the traditional approaches, the system will be scalable and proactive. The explicit nature of agent creation and communication allows user to easily understand what is going on the system, and we can simply create new types of agents to increase functionality of system, not need redesigning the whole system. A more traditional Internet solution to this problem would be to simply create a web page of the Airline, just to list all the flight information. Mobile agent technique deliver a new paradigm for Internet computation.

5. REFERENCE

[1] Yu M, You J.Y. "Component migration mechanism for load balancing in mobile agent system". *Computer Engineering*, 23(12):12-14 2001. (In Chinese).
[2] Bic L F., Fukuda M., Dillencourt M B. "MESSENGERS: Distributed Programming Using Mobile Autonomous Objects", *Journal of Information Sciences*, 1998
[3] Green R, Pant S. "Multi-agent data collection in Lycos", *Communications of the ACM*, 42(3):70-75, 1999.

Research on Complex Biological Systems with Automata Network Based on the Evolution Technology of Bisection*

Zhongjun Wang[1][2] Nengchao Wang[1] Xingqin Cao[1]
Parallel Computing Institute, Huazhong University of Science and Technology
Wuhan 430071, P.R.china[1]
Department of statistics, Wuhan University of Technology
Wuhan 430063, P.R.China[2]
E-mail: Wang_zh3000@sina.com

ABSTRACT

Biological system is a complex evolutionary system, and the evolution of biological system result from inheritance and mutation's bisection evolution. The automata network is a effective tool for simulating a complex biological system. Bisection evolutionary technology simplity the evolving process, and help for automata network model. This paper discusses the complex biological system with automata network method based on the evolution technology of bisection.

Keywords: complex system; automata network; evolution

1. INTRODUCTION

A new science has emerged in recent years in an attempt to explain the inner "workings" of complex systems. Here "complex" does not mean "complicated", but rather something like "full of diverse, intricate, interacting structures".

Complex system research is a multidisciplinary field covering the gamut from Physics, Mathematics, Computer Science, Biology to Economics. Whilst there is no good agreement on what the term "Complexity" might mean, there are a number of threads or concepts linking the different disciplines together into a coherent field of research. Complex systems typically do not fit within the confines of one of the traditional disciplines, but for their successful study require knowledge and techniques from several disciplines. Examples of systems studied include: ecology, evolution, economics, artificial life, genetic algorithms, neural networks, cellular automata.

The idea of what is a complex system is not well defined, however people intuitively feel that a biological system is complex. This paper discusses the complex biological systems with automata network based on the evolution technology of bisection.

2. COMPLEX BIOLOGICAL SYSTEMs

Biological system is a typical complex system. It takes on several basic characters of complex system.

Hierarchy
The biological systems have a hierarchy and interactions between objects at high level are very well defined. Evolutive phenomena of each hierarchy are different, and the rule of development exists diversity. For example, cells in the human body have the same basic structure with a cell wall, cytoplasm, nucleus, DNA, RNA etc. Individual cells have further refinements to the basic structure to perform the specific function assigned to them.

Coupling Architecture
Very few objects in a sub-system should interact with objects in the other sub-system in the biological system. They form a network and have a strong relevancy, interaction and coupling between different parts, difference layers, even the same parts and layers.

Non-linear
The interaction is non-linear between parts and layers in the biological system. This is one of the cause to produce complex and diversity.

Evolution
Complex biological system is a evolution system. Its structure and function are dynamic. The core problem of complex system is its evolutive behavior.

Open Architecture
Complex biological system is a open system. It exchange substance, energy and information by circumstance.

Intelligence Architecture
All kinds of parts will spontaneously form structure and phenomnena of time and space order in the course of evolution.

Consequently, complex biological system is a important research area of complexity science.

Biological system's development is a gradually evolution process from low to high, from simple to complex. Darwin's theory of evolution, by means of natural selection, had proved the evolution thought by a large of fact. All biological systems have evolved over a millions of years, with each generation evolving to better adjust to the changing environment. Most changes take place gradually but sometimes there were quite sudden and major changes caused by mutations.

Complex biological system is a evolution system. We will use bisection evolution technology to study the evolution system.

3. THE EVOLUTION TECHNOLOGY OF BISECTION

The evolution technology of bisection is a basic technology to study the complex biological system.

* Research supported by the 863 plan of China (2001AA231071) and the Natural Science Foundation of China (NSFC Grant No.60173046).

Complex biological systems are formed by twelve billion years. The process is very long from C, H, O and N element to organism. The long evolution process is a bisection evolution process. I try to describe the bisection evolution process by figure 1.

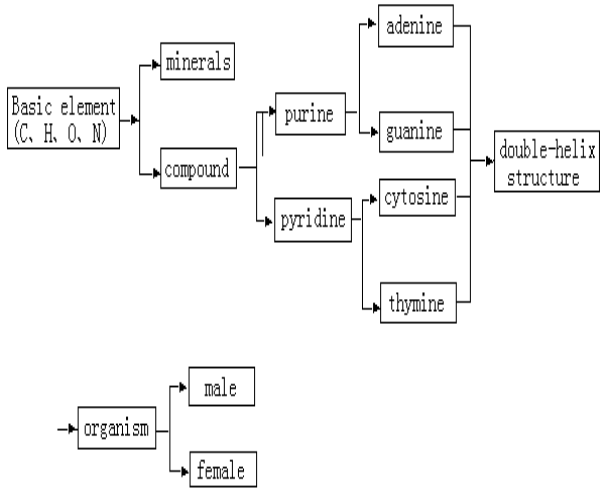


Figure 1 Shows biological bisection

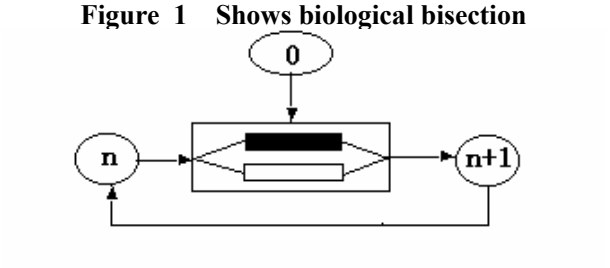


Figure 2 Bisection evolution dynamic system

The bisection evolution model can be described as discrete dynamical system as figure 2

The bisection evolution dynamic system is set up by professor Wang Nengchao in middle of 90's. In figuer 2, '0' is evolutive initial state, 'n' is evolutive n step's state, ■ and □is evolutive rule of mutual opposition and unification. This kind of state evolution is known as bisection model. Each process includes two parts—dispartment and combination, the first is to dispart two opposite states from the old state, and then combine the two states to produce a new state. This process from dispartment to combination made a matter evolution from a state to another.

According to Darwin's theory of evolution, two mainly potences are important in the organism evolution process. These are inheritance and mutation. Thus we can use bisection evolution model to review the complex biological system as figure 3.

Figure 3 shows that biological system is forever evolving. The evolution result in the development of species from low to high, from simple to complex. Inheritance and mutation is two opposite character in the evolution process. Inheritance can keep species stable existence. Mutation is everything common character. It can provide motivity and create condition for evolution.

Complex biological system is a dynamic evolution system. The evolution of biological system is result of inheritance and mutation's bisection evolution. Like this, we can understand the biological system more deep, and describe it more simple.

Because complex biological system has hierarchy, coupling architecture, open architecture, evolution and so on, the modeling according to simple rule will has more practical significance. Based on the bisection evolution technology, we study and simulate the complex biological system, at present, automata network method is a effective, and may be unique.

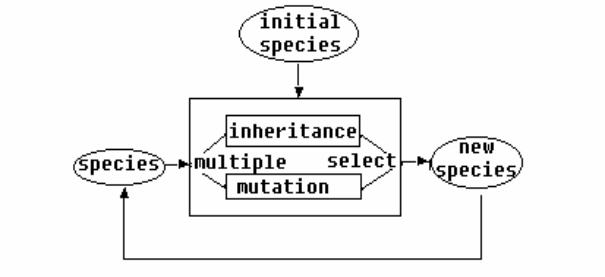


Figure 3 Bisection evlution model of biological system

4. AUTOMATA NETWORK MODEL

Automata network model presents a universal framework of complex system. It possesses definite validity and operation, and is closer to real system. At present, automata network method is a apt choice to study biological system.

When automata network method simulates complex biological system, cells are considered as regular lattice. For example, simulating biological configuration structure and creature behavior of individual, cells always vary by time.

In automata network, each unit cell can be different, or a network system is consist of several sub-networks of automata. For example, when simulating the network of different cells or individuals, we must use different unit to form automata network.

The state change function of network structure and unit automata can vary by time. This is because biological system is evolving complex self-adapt system, organism can adjust itself behavior according circumstance change, namely feedback mechanism.

Studying automata network of complex biological system, key is to build a fit automata network model. Biological mechanism and bisection evolution technology are important criterion. Thought automata network uses simpler model whether system units or between units. But biological entities never align, and interact rarely only according invariable laws and neighbors. So modeling in complex biological system, not only need enough simple so as to expose its essence, but must think organism's particularity for fear losing its important attribute in the process of modeling.

5. CONCLUSIONS

Biological system is a complex evolution, and its evolving mechanism is bisection evolution. The automata network is a effective tool for simulating a complex biological system. Bisection evolution technology simplity the evolving process, and help for automata network model. Automata network is a perhaps the clearest example application of parallel computing. As the interactions are between nearest neighbours, and are typically homogeneous, traditional data parallel techniques can be used to great effect. Even inhomogenous units can be efficiently computed with MIMD computational model.

6. REFERENCES

[1] S.Wolfram, A New Kind of Science, 2002 by Stephen Wolfram, LLC
[2] Burrows MT, Hawkins SJ，Modelling patch dynamics on rocky shores using deterministic cellular automata，MAR ECOL-PROG SER 167: 1-13 1998
[3] Viher B, Dobnikar A, Zazula D，Cellular automata and follicle recognition problem and possibilities of using cellular automata for image recognition purposes，INT J MED INFORM 49: (2) 231-241 APR 1998
[4] Verburg PH, de Koning GHJ, Kok K, et al.，A spatial explicit allocation procedure for modelling the pattern of land use change based upon actual land use，ECOL MODEL 116: (1) 45-61 MAR 1 1999
[5] Hill MF, Caswell H，Habitat fragmentation and extinction thresholds on fractal landscapes，ECOL LETT 2: (2) 121-127 MAR 1999
[6] Dunkerley DL，Banded chenopod shrublands of arid Australia: modelling responses to interannual rainfall variability with cellular automata，ECOL MODEL 121: (2-3) 127-138 SEP 15 1999
[7] G. Weishuch, Complex systems Dynamics: an introduction to automata networks, 1991 by Addison-Wesley Publishing Company
[8] 郭卫斌，王能超，施保昌. 复杂生物系统及其二分演化机制 自然杂志，2001，23（3），p172-176.
[9] 王能超. 同步并行算法设计的二分技术. 中国科学（A辑），1995，25（2）
[10] 王能超著.同步并行算法设计. 科学出版社，1996
[11] 王能超. 千古绝技“割圆术”（Ⅰ）（Ⅱ）。数学的实践与认识，1996，26（4）：315~321；1997，27（3）：p275~280.
[12] 王能超. Walsh 函数的数学美（Ⅰ）（Ⅱ）. 云南大学学报，1997，19（增刊）：p306~323.

Connecting Distributed Fieldbus Networks to Ethernet

Jianbin Zheng
College of Information Engineering, Wuhan University of Technology
Wuhan 430070, P.R.China
E-mail: Zhengjb@public.wh.hb.cn
and
Guanxi Zhu
Department of Electronics and information Engineering, Huazhong University of Science and Technology
Wuhan 430074, P.R.China
E-mail: gxzhu@public.wh.hb.cn

ABSTRACT

Important reasons for connecting fieldbus networks to IP-based networks are the provision of remote access for monitoring and maintenance purposes, but also the inclusion of automation systems into an enterprise-wide management scope. The model of interconnection between fieldbus networks and Ethernet is presented in the paper. The communications between managers and agents are based on Simple Network Management Protocol (SNMP). We propose a modular structure of agent, with a master agent performing all SNMP-related functions, whereas the subagent responsible for getting and setting network variables on fieldbus. All information about fieldbus is stored in a tree-like structure called a MIB. By means of prototype implementations of agents for different fieldbus systems (Profibus, LonWorks and CAN), we study the influence of the underlying fieldbus communication principles on the implementation and operation.

Keywords: Fieldbus, Ethernet, Interconnection, Manager, Agent

1. INTRODUCTION

Many different kinds of fieldbus systems, ranging from very small and primitive networks to more sophisticated networks, have been developed to meet the requirements of industrial and also building automation. Despite international standardization efforts, a large variety still exists and will remain because of being tailored to distinct applications and with different goals. This reflects the wide diversity of applications that are served by fieldbuses.

Profibus-DP [1], which means “Process Fieldbus for Decentralized Peripherals”, focuses on sensor/actuator communication applications. The development of Profibus was funded by the German government during the period of 1987 and 1990 as a cooperative project between the universities and private enterprises.

LonWorks [2], which stands for “Local Operating Network” , was developed at the end of the 1980s by Echelon, US. A family of chips is available as standard products from Motorola and Toshiba. Mainly are applied in building automation.

CAN [3], acronym for “Controller Area Network”, originally had been invented and driven by Bosch, Germany, at the beginning of the 1980s. Now it has been designed into many areas related to vehicles and industrial control, such as: passenger cars, robot control, laboratory automation.

A common restriction of all fieldbuses is their limited extension. E.g., Profibus only allows a maximum extension of 800m at 500kbit/s data rate [1]. This restriction causes the existence of fieldbus networks as isolated islands, especially in larger installations. All fieldbus systems have to be configured and maintained separately, and communication between those islands is impossible.

From a management point of view, a unified appearance of field area networks (FANs, fieldbus system) is desirable for the user who is in most cases not interested in the fieldbus per se but rather in the actual application. It is important to hide the differences between the fieldbus systems and make different FANs look the same, on an abstract level.

In the recent past, the breakthrough of the Internet as a worldwide accepted communication medium has stimulated much work in the area of fieldbus/Internet connections. Ethernet especially for internet connections, Ethernet interfaces become increasingly popular.

Today, the interconnection of FANs and the Ethernet is one of the most challenging topics in automation. The benefits of connection between FAN and LAN are evident: direct integration of fieldbus data into comprehensive, well-known data acquisition systems, remote access for monitoring and maintenance purposes, as well as the inclusion of automation systems into an enterprise-wide management scope. Finally, with the advent of the Internet, an interconnection on IP level promises seamless integration into worldwide corporate or public networks with virtually unlimited range and interesting application in industrial as well as building or home automation [4,5].

There is, of course, a severe restriction for the interconnection between fieldbus systems and a LAN. Its feasibility is confined to applications where both fixed data rate and predefined response time are not of predominant interest. The reasons are simply the lack of real-time behavior in IP-based networks and sometimes the unpredictable availability of connections. An obstacle to the practical integration of different fieldbus systems into one homogeneous framework is the still existing large variety of incompatible FANs. Nevertheless, from the view of network management, integrated solutions are available.
The main objective of the work presented here is to provide a modular and extendible gateway between FANs and Ethernet. The ultimate goal is the integration into a company-wide comprehensive management that has to handle different systems like Profibus, LonWorks, CAN. Hence, we focus on a top-level view for individual FANs. The model of interconnection between FANS and Ethernet, master agent (MA), sub agent (SA) and their relationships are shown in Fig.1.

One motivation for the realization of an agent is the separation of the protocol processing done by the master agent and the processing of MIB variables done by the sub agent. Another important task performed by the MA is the merging and administration of the data gathered by the sub agents. As a result, the SAs are not bothered with SNMP protocol details and MIB implementers can concentrate on their proper tasks. On the other hand, the SAs perform all the low-level tasks concerning the FAN control, like getting and setting network variables (NVs) or initiating actions on FAN elements. Furthermore, the mapping of FAN variables to FAN objects is done by SA.

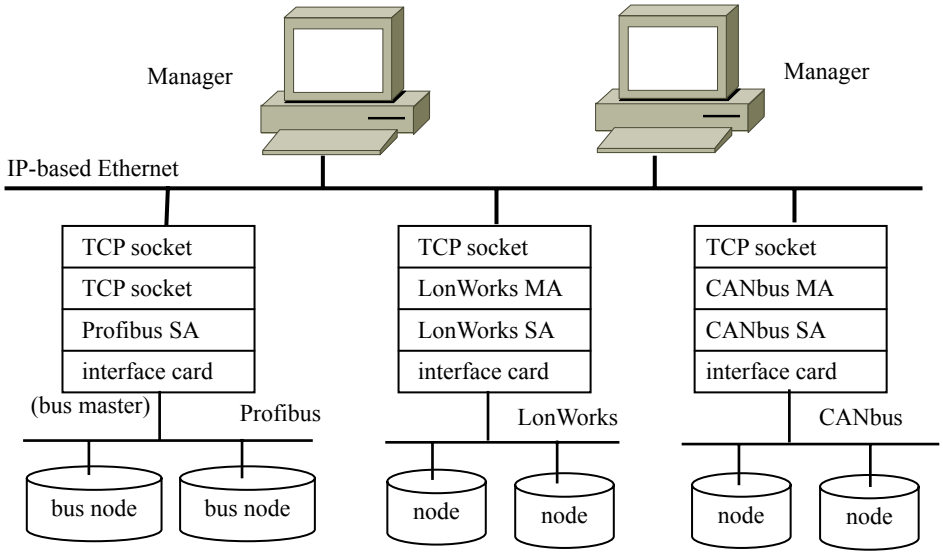


Fig.1 Model of interconnection

2. SNMP AND MIB [6]

When talking about fieldbus-Ethernet interconnection, we understand that the connection is achieved by agents. These agents are collecting and storing information about the state of their network elements in a management information base (MIB). Until now, Simple Network Management Protocol (SNMP) is the only protocol for element and network management that has achieved broad acceptance among many proprietary solutions [7].

Due to simplicity and low resource consumption of SNMP at the managed device, it is still the protocol of choice for the majority of LAN devices to provide at least some status information. So far, three versions of SNMP have been defined. In the following, we will restrict our considerations to the most widely used first version SNMPv1. From today's view, even if SNMP might not be the best protocol for all situations, it is the one with the largest number of readily available tools and application programming interfaces (APIs).

SNMP defines a clear client/server relationship between devices. The manager communicates with several agents to retrieve data or set new values on the managed device. All accessible data at the agents are stored in a MIB. SNMP only describes how to represent management information and how to transfer it.

SNMP is an application-layer protocol, designed for using the connectionless service User Datagram Protocol (UDP) as transport layer. Only five protocol data units (PDUs) are defined for the communication between manager and agent (Fig.2): GetRequest, GetNextRequest, SetRequest, Trap and GetResponse. The loss of a PDU is usually handled by the manager application by means of timeout and retry mechanism. A Trap is sent by the agent (very much like an Interrupt) to report an important situation to the manager which then has the possibility to immediately react upon this information.

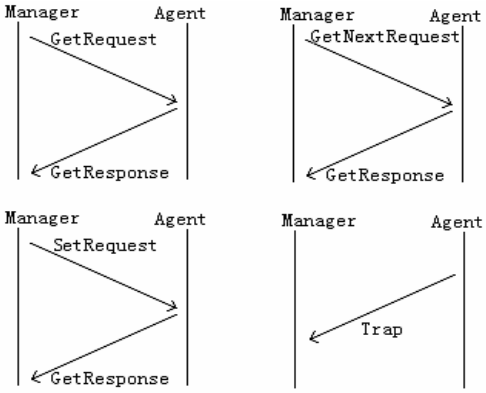


Fig.2 SNMP protocol operations

All information at the agent is stored in a tree-like structure called a MIB. Its contents are described in the notation of ASN.1. All data that can be addressed and manipulated by management applications must be part of the MIB. Each element of the MIB is unambiguously identified by assigning it a globally unique Object Identifier(OID), a sequence of nonnegative integers delimited by periods. In addition, a textual name is associated with each OID. One important feature of the SNMP MIBs is that they are defined in a worldwide unique name space. Several sub-tree of this name space are standardized MIBs, whereas others can be reserved, for example, by companies to add individual extensions. One of the most valuable aspects of SNMP is the plethora of MIBs containing thousands of OIDs already standardized and ready for use.

3. MODULAR AGENT

In the original SNMP standards, only the functionality of manager and agent devices was defined. This quite abstract

view completely hides the details of an actual implementation-the agent only acts as some sort of dumb data container for information about a network node, seen as a monolithic piece of soft ware doing all the processing.

For our purpose of connecting FANs to an IP-based network, such a modular design promises advantages for the implementation. A logical decomposition is to have an MA taking care of the connection to Ethernet and have a SA providing the low-level attachment to specific FAN each. Hence, it seemed reasonable to store all MIB data together in the MA, while the SA is kept as simple as possible and just provide a unified interface to the fieldbus. MA and SA are placed on the same machine. Using this assumption, an interprocess communication mechanism like shared memory could serve the demands quite well. The structure of MA and SA are shown in Fig.3 and Fig.4, respectively.

Since a FAN normally consists of a fair amount of nodes, making each of them directly addressable by the management application would result in an unwanted complexity together with the need for a huge amount of IP addresses. In our approach, one gateway represents several fieldbus nodes of one or more FANs within one single SNMP MIB. As shown in the previous paragraphy, a modular structure of the agent is reasonable, with a MA performing all SNMP-related functions, whereas the SAs are responsible for getting and setting network variables or initiating actions on FAN elements.

As most of the interfaces to FAN systems are available on PCs using some sort of 32-bit Windows (Win9x, WinNT, Win2000), a Windows-specific mechanism (DDE, OLE, or OPC) could be used. But taking other operating systems into account and because of their availability on nearly every system and their low resource consumption, a socket communication seems far more versatile.

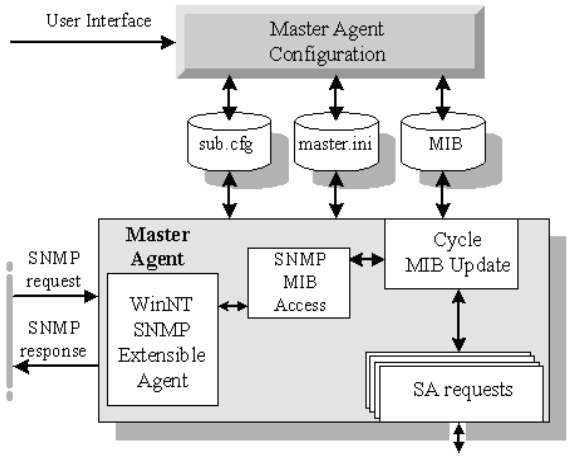


Fig.3 Structure of MA

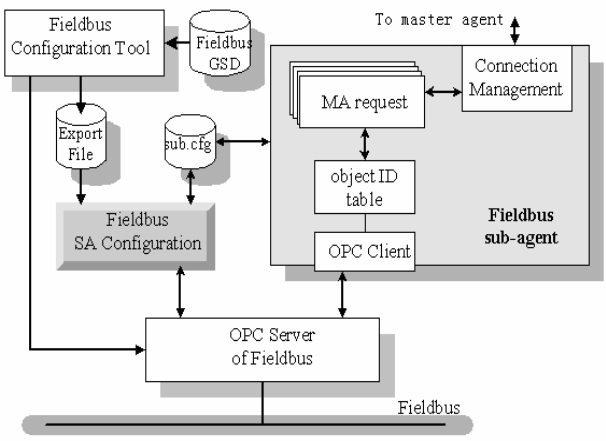


Fig.4 Structure of SA

An important aspect of the implementation is the way how data are organized in the agent and how the MIB variables are updated. In particular, the selection of the data offered to the manager deserves careful consideration. Simply passing all available information about the connected fieldbus to the agent would be a rather crude solution with two major shortcomings. First, the amount of data may become overwhelming for complex FANs, although much of the information may not be relevant for the manager. Second, some a priori knowledge about the actual setup of the FAN might be necessary. One 8-bit input port, for instance, can represent eight different switches. If just the value of the entire port is transferred, the manager application would be required to know the meaning of each bit in this value. In addition, the selection of the data made available to the agent already provides a basic level of security, since only these data are represented in the MIB and can be accessed.

As a first step, all available FAN variables are presented to the user to distill the desired objects from this information. By the term fieldbus object, we understand in this context any meaningful piece of information about either the FAN itself or its environment, regardless of its physical representation on the fieldbus. The term fieldbus variable, by contrast, denotes a coded entity of data that can be transferred over the fieldbus network at a time. Thus, a variable may contain one or more objects. The information on how to extract objects from variables is stored within the SA and must be furnished by the user. At this level, we have already achieved a fieldbus-independent and abstract interface for data representation.

The second step is the configuration of the MA with a FAN-independent tool. At this point, the SA sends a list with its SA objects to the MA, and the user selects the relevant SA objects for each MA. These SA objects are then placed in the SNMP MIB of the respective MA and are assigned an object identifier.

For normal operation, we favor an implementation where the agent acts as caching device to minimize communication delays between manager and MA and to reduce data traffic to and from the SA. Therefore, each MIB variable has a refresh interval (specified in milliseconds) and the MA autonomously and cyclically collects the data from its SA. The SA is responsible for gathering the data and handing it over to the MA. How the SA fetches the data from the FAN is implementation specific and depends both on the properties of

the fieldbus and the capabilities of the interface used.

4. PROTOTYPE IMPLEMENTATIONS

In this section, we briefly describe implementations designed to validate the practicability of the suggested approach. In addition, they show which difficulties may arise for several FANs. The examples show that the structure of the SA varies grossly. The MA, by contrast, always remained the same. For all prototypes, an MS Windows'2000 Professional platform was used.

The two main tasks of the actual MA are to handle SNMP requests and to keep the contents of the MIB up to date. The MA was implemented using the SNMP agent provided by Windows. This agent is extendible by user-defined Dynamic Link Libraries (DLLs), each handling specific MIB subtrees. During configuration, the MA configurator searches for active SA on a predefined port to establish a connection. Subsequently, it requests the list of subagent objects the SA can offer. Once the list has been retrieved, the user has to select the objects that are to be included in the SNMP MIB of the MA. This selection is stored in a configuration file and subsequently used to initialize the MA, which then runs entirely without direct user interaction.

Profibus-DP is a single-master system. This fact imposes additional restrictions on the design of the SA. First of all, the SA cannot be simply attached to the bus as an additional node. Consequently, important information such as device configurations of certain diagnosis data could neither be read nor changed by the SA. The second constraint is that the data flow on the bus is controlled by the bus master. Therefore, the data exchange between the SA and individual bus nodes should also be handled by the master. In typical implementations, the cyclic data are stored in a well-known shared memory area that can be accessed by both the bus communication module and the operating system of the PC. A rather intricate point is the handling of acyclic data transfer. This information has to be added manually in the SA configurator, regardless of the approach used.

In LonWorks, the overall network may consist of nodes, subnets and domains. The communication principle is based on the exchange of variables, as they are known from standard programming languages. The user does not see the actual communication, he simply accesses a variable through read or write commands in his application program. A network installation tool, such as the LonMaker Integration Tool (Ver3.0), is generally used to meet the requirements of installation and maintenance. The configuration of network and MIB of network variables are established after installation. NodeBuilder (Ver3.0) include network-variables explorer, which can explore any network variable in any node and modify any input network variable. LNS DDE server (Ver1.5) is based on Windows, so any user can conveniently develop application program of client.

CAN is a system based on network messages, where the sender transmits its information without knowing whether there is anybody interested in it. Any receiver can use the data provided the meaning is known, which also requires configuration. In such systems, it is very easy for a management agent to gather information. When the bus is free any unit may start to transmit a message. The unit with the message to be transmitted in highest priority gains bus access.

5. CONCLUSIONS

We have shown that the standardized protocol SNMP is an optimal solution for connecting fieldbuses into Ethernet from today's point of view. Fieldbus devices can be made accessible from SNMP managers by the use of an agent. The prototype implementations designed validate the practicability of the suggested approach.

6. REFERENCES

[1] German standardization office, DIN 19245- Profibus Specification.
[2] Introduction to the LonWorks system, http://www.echelon.com/support/documentation
[3] CAN, Controller Area Network, ISO/DIS 11898, 11519-1.
[4] M. Wollschlaeger, "Mapping of fieldbus components to WWW-based management solutions", Fieldbus Technology, Berlin, Germany: Springer Verlag, 1999, pp.172-179.
[5] D. Dietrich, T. Sauter, "Evolution potentials for fieldbus systems", Proc. IEEE Int. Workshop Factory Communication Systems, Porto, Portugal, Sept.6-8, 2000, pp.343-350.
[6] D.T. Perkins, Understanding SNMP MIBs, Englewood Cliffs, NJ: Prentice-Hall, 1996.
[7] M. Kunes, T. Sauter. "Fieldbus-Internet connectivity: The SNMP Approach", IEEE Trans. on Industrial Electonics, Vol.48, No.6, Dec, 2001, pp.1248-1256.

Constructing information service platform for the Digital Basin using the web service

Zhou Xiaofeng Wang Zhijian Ai Ping Li Shijin
(School of Computer & Information Engineering, Hohai University, Nanjing 210098)
E-mail: z_xiaofeng@sina.com

ABSTRACT

The Digital Basin comes from the Digital Earth, which is a huge information system with heterogeneous and distributed components. It is the key for the Digital Basin to solve the problems of information sharing, the functions reuse and the distributed object cooperation. The web service has both advantage of web which provides cooperation of objects on different platform and feature of component which provides function reuse, so that it provides feasible technology for solving aforesaid problems. We discuss the framework, the applying project and the technology features of constructing information service platform for the Digital Basin using the web service, which offers reference for success of the Digital Basin.

Keywords: Digital Basin, web service, platform, distributed systems, framework

1. INTRODUCTION TO THE WEB SERVICE

Web service is an extension of object/component technology in Internet, it is a object/component technology installed on Web. The web service has both stand-out capability of developing module based components and web. First, the web service has the function of black-box liking component, can be reused under the circumstance not caring for how the function is achieved; At the same time, the web service differing with traditional component technology, provides cooperation between them through integrate different types function modules opened out different platform. So web service is defined as a model of next generation distributed systems development used widely.

Now there is not an accordant definition for web service. Microsoft's definition is: XML Web services are units of application logic providing data and services to other applications. Applications access XML Web services via standard Web protocols and data formats such as HTTP, XML, and SOAP, independent of how each XML Web service is implemented. XML Web services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model[1]; IBM's definition is: Web Services are self-contained, modular applications that can be described, published, located, and invoked over a network, generally, the Web[2]; W3C's definition is: A Web service is a software application identified by a URI, whose interfaces and binding are capable of being defined, described and discovered by XML artifacts and supports direct interactions with other software applications using XML based messages via internet-based protocols[3].

Web service has the characteristic as follows:

Encapsulation: because web service is an extension of object/component technology, so it has good encapsulation like object/component;

Cooperation: Using SOAP protocol, any web services can interact with other web services, in spite of these web services based on what developing environments are and are realized in any program languages;

Standard: The web service uses open standard protocols to describe, transport and exchange information. These standard protocols are provided with complete free criterion in order for any squares to realize them.

Integration: because the web service take simple intelligible standard web protocol for description of component interface and criterion of co-description, completely shield difference of different software platforms, so different web services can be highly integrated;

Simplicity: the protocol used by the web service is simple, and supported by a lot of free tools, establishment and deployment of web service are simple, it is easy to transfer traditional object/component to web service.

Now, the web service technology has given broad self-identity. W3C is instituting correlative standards. Documents produced by W3C include SOAP version 1.2 and WSDL Version 1.2 etc. Web Service "stack" is shown in Figure 1:

This shows that the protocols related the web service include Simple Object Access Protocol(SOAP), Web Services Description Language(WSDL), Universal Discovery Description and Integration(UDDI), Web Services Flow Language(WSFL). Many of higher layer protocol wait for opened up, such as routing, reliability and transaction protocol.

???	???
Routing, Reliability and Transaction	???
Workflow	WSFL
Service Discovery, Integration	UDDI
Service Description	WSDL
Messaging	SOAP
Transport	HTTP, FTP, SMTP
Internet	Ipv4, Ipv6

Figure 1 Web Service "stack"

Web service Architecture consists of 3 stacks, that are closely related. A transport stack is for standards that are exchanged on the wire. Description is for describing an individual or collection of services. Discovery is the finding of services.

2. INTRODUCTION TO THE DIGITAL BASIN

2.1 Definition and implication of the Digital Basin

The concept of Digital Basin explicates from Digital Earth. It is an information aggregation about the basin on Digital Earth. Further on, the Digital Basin is an organic whole that digitize all the information of the basin and its interrelated, and structure by using form of the spatial information. Thereby it effectively reflects the integrated and true situation of the

whole basin from each side, and fills various needs of transferring information.

The Digital Basin is a system platform inosculating all kind of digital information within basin based on basin spatial information, it is a uniform and digitized re-appearances of true basin and the phenomenon related it, it moves basin into lab and computer, then becomes virtual counterpart of true basin. The Digital Basin consists of database included various information and sub-system of data gathering, processing, exchanging and managing, can make comparison and analysis of data of different period based on different needs, and penetrates its variational rule.

The primary purpose of the Digital Basin is to provide uniform effective reliable access platform for multi-source heterogeneous huge data share, to solve existent problems now for instance difficulty of data share, lack of efficiency of software programming, lack of degree of software standardization, difficulty of ensuring software quality and difficulty of system integration. Through gathering and digital managing various information of full basin about geography condition, natural resource, environment, humane sight, society state and economy state, constructing integration information platform and 3-D image model of full basin, all levels departments can effectively manage economy construction of full basin and make macroscopical policy of resource used and developed.

2.2 Architecture of the Digital Basin

The Digital Basin consists of three parts generally. First is the digital infrastructure, secondly is the information service platform, thirdly is the application. The three parts form an

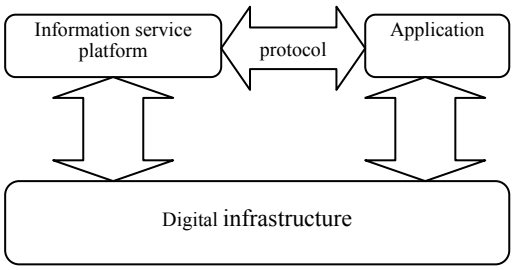


Figure 2 system general architecture

organic whole through protocols and interfaces. Its general architecture is shown in Figure 2.

The digital infrastructure consists of all data gathering, network and hardware, it is a basic resource of the Digital Basin. The operation applications is a functions representation of the Digital Basin, developing different operation applications fills different operation needs of managing and harnessing basin. The information service platform is the core of the Digital Basin, it is the administrant of resources and a provider of services all of the Digital Basin, the information service platform realizes sharing of information and services within network through normal mechanism of distributed object interoperation, and agglomerates the Digital Basin.

2.3 Features of the Digital Basin

The Digital Basin is the application of information technology in basin operations, so essentially say, is an information system. But it exists remarkable characteristic listing below comparing with generic information system.

Distributing: the Digital Basin is a typical distributed system. Because the basin department has characteristic of distributed on clime and uses grade management mechanism, the information called by the Digital Basin is distributed, the operation applications calling information is distributed too.

Opening: the Digital Basin is not an isolated system, is organic content of the nation informatization, it need support by outside information and provides information services for outside too, so it is an open system.

Complexity: the Digital Basin is a organic whole consisted of thousands upon thousands operation application systems and its hugeness support environment, has complicated relation with each operation application systems, operation application systems and support environment, simultaneity the Digital Basin needs support the operation applications and users on multi-lay, so the Digital Basin is a complicated system.

Hugeness: the information disposed by the Digital Basin is huge, the number of operation applications included by the Digital Basin is immense, the user group of the Digital Basin is giant, so the Digital Basin is a huge system.

Heterogeneity: the data disposed by the Digital Basin is multi-source, multidimensional and heterogeneous, this requires the Digital Basin has the capability of shielding difference of different data, the platform and system faced the Digital Basin is heterogeneous, this requires the Digital Basin has the capability of interoperating among heterogeneous environment.

Wholeness: the Digital Basin is a isolated whole, moreover does not load with a lot of intersected data and operation application systems. The wholeness of the Digital Basin is a important pledge realizing data shared and reducing redevelopment of the operation application system.

Timeliness: the contents included in the Digital Basin is not changeless, it is necessity to add the contents of the Digital Basin or reduce along with change of operations and development of technologies, so this claims that the Digital Basin has good expansibility.

3. CONSTRUCTING INFORMATION SERVICE PLATFORM FOR THE DIGITAL BASIN USING THE WEB SERVICE

3.1 Platform framework

For its excellent interoperability and integrity, Web service is especially suitable for constructing the information service platform for the Digital Basin, which is distributed, heterogeneous and complicated. Following is the architecture of the platform.

In the platform, the storage facility is for the various information needed by the Digital Basin, including various kinds of databases, knowledge database, and etc; the platform manager is responsible for the coordination of the components of the platform, which is mainly providing directory service, information deployment, load balancing, configuration, authentication, accounting and priority, etc. The content of this part is independent of specific service and application. The main body of the Digital Basin includes objects, components, middleware, functions, and subroutines, which implement the functional logic of digital basin. And These also can be on different platforms and implemented in different languages. To do so, These modules must firstly be

described by WSDL and registered and deployed in the registration center using UDDI. Web Service is the provider of information service and if required, there can be multiple web services in the Digital Basin, which can provide different kinds of information services and can be interoperated through SOAP.

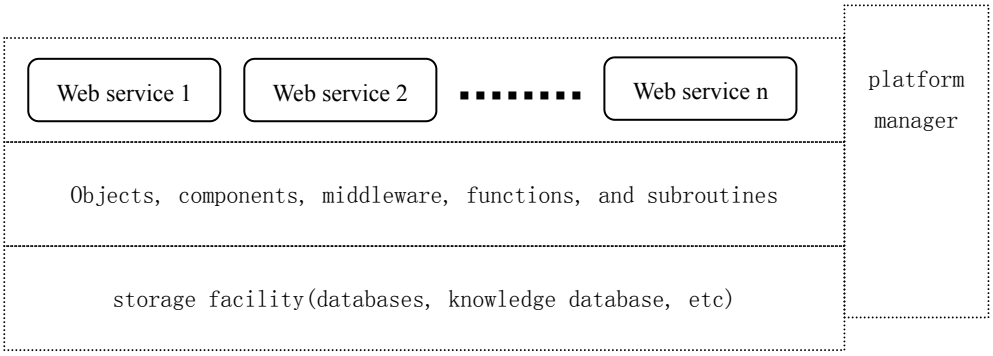


Figure 3 the architecture of the platform of the Digital Basin

The informationservice platform is in logic, while it is distributed physically, which is responsible for providing the related services and management of resources. The components of the platform may be constructed on different systems and implemented in different languages. And it can not be developed isolately and will be upgraded with the new requirements of the application. In another words, the platform is constructed by the requirements of the application in accordance with the standards of WEB service.

3.2 The implementation of the application system

The Digital Basin information service platform based on the web service, must be constructed in accordance with the standards of web service. Following is the flowchart of the process:

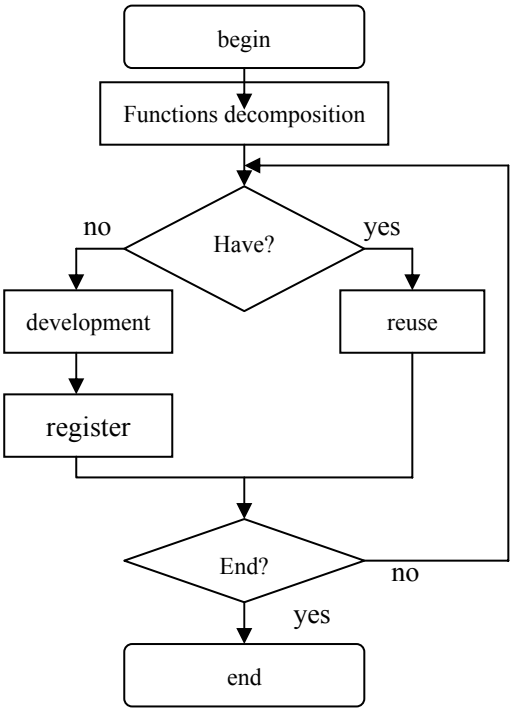


figure 4 development flow chart of the application

While developing the single application system, it should be decomposed into small independent functional modules firstly. Then, Queries should be conducted to find whether these modules have been implemented. If so, software reuse is the next step. Otherwise, they should be developed from the scratch and registered into the platform and become an integrated part of the system. After all the modules are finished, so is the entire system.

3.3 Feature of the platform

When constructing the information service platform for the Digital Basin based on the web service and developing the operation applications systems, we can interoperate between different systems and integrate the huge systems, which can also make good use of information sharing and reuse of functions. The main advantages are as following:

Interoperability between different objects: Web service can greatly support the interoperation between different objects and the user can decide to use which programming language under some specific conditions. And the implemented modules can be parts of the platform. In another words, the user can obtain all the services provided by the information service platform, with no regard to whatever language it is implemented and in whatever environments.

Scalability: On one hand, heterogeneous environments can be integrated into the platform, and on the other hand, new functional modules can also be added into it to extend and upgrade the system, as there are the requirements.

Maximum reuse: All the functional modules are the services provided by the platform, and any valid users can access these services. When a new functional module is required, the user can query the platform and if it is existent, the existing one can be reused. Otherwise, the user needs to develop it.

Transient access to the data: what the platform provides are services. And the user can access the data resources through the services provided by the platform, while the user needn't to know the location and format of the needed data, i.e., the access is transient.

Inheritance to the legacy system: As the functional logic of the platform may be implemented as objects, middleware and functions et al, the legacy system can be transformed into a suitable form of functional modules, i.e., the object-oriented technology is not required.

4. CONCLUSIONS

Currently, the standardization of the web service has not been finished yet. However, the main companies of IT have disclaimed to support it and have issued their products, such as the .NET of Microsoft, WebSpher of IBM, and SUN ONE. The information service platform can be constructed on these new technologies. The Digital basin is a very very complex huge system, and there are no examples that can be followed. We have designed the DIGITAL YELLOW RIVER project by

using the web service technology as the core, which have tackled the problem of the information sharing and system integration. The implementations are underway.

5. REFERENCES

[1] .NET Glossary，
[2] http://www.microsoft.com/net/defined/glossary.asp
[3] IBM Web Services Architecture team, Web Services architecture overview
[4] Web Services Architecture Requirements,
[5] http://www.w3.org/TR/2002/WD-wsa-reqs-20020429#N100CB
[6] cai xiaolu, constructing Web Service,
[7] http://www-900.ibm.com/developerWorks/
[8] SOAP Version 1.2 Part 1: Messaging Framework,
[9] http://www.w3.org/2000/xp/Group/
[10] Web Services Description Language (WSDL) 1.1,
[11] http://www.w3.org/TR/2001/NOTE-wsdl-20010315
[12] UDDI Technical White Paper,
[13] http://www.uddi-china.org/pubs/UDDI_Technical_White_Paper.pdf
[14] Web Services Flow Language (WSFL 1.0),
[15] http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf
[16] Al Gore, The Digital Earth: Understanding our planet in the 21st Century at the California Science Center, Los Angeles, California, on January 31, 1998

An Automobile License Recognition system Based on Neural Networks

Liu Kewen Xiong Haoyu
The School of Information Engineering, Wuhan University of Technology
Wuhan, Hubei 430070, China
E-mail: xionghx1976@21cn.com

ABSTRACT

An automobile license recognition system scheme based on Hamming NN is put forward. The segmentation way of automobile license realm location, the extraction technology of intelligent character, the template matching NN that can recall subnet, and character recognition system scheme with hierarchical hybrid integrate classifier are discussed in this paper.

Keywords: Hamming Neural Networks, License Recognition, Image Segmentation, String Location.

1. PREFACE

In recent years, with the rapid development of transportation modernization, license recognition has now become one of important study schemes in intelligent transportation in computer vision and mode recognition technique realm. License automatic recognition technique is paid more and more attention by people. License location, character segmentation, character automatic recognition and subsequent process are the key technologies in license automatic recognition technology. If we can use computer to recognize license, we can monitor cars automatically and have no need to add other special devices in cars, and bring great convenience for the automatic management of transportation system. So license automatic recognition system is one of technologies to push forward the computerization of transportation management.

2. INTRODUCTION OF THE SYSTEM

The vehicle license automatic recognition system comprises of four modules: Image gathering module, License location module, Image process, recognition module & Block communication module (Figure 1).

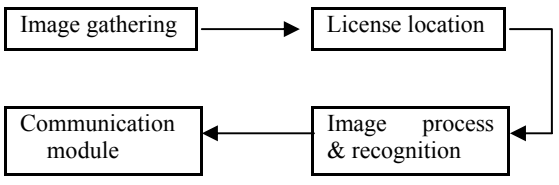


Figure 1 Map of license recognition system

When a car passes the platform, The computer can detect the car's arrival signal, and inform the microcomputer to gather the car's rear image, at least 6 to 8 images every car, only in this way can we ensure at least having an image which contains an intact car number. Then the system will process the segmentation of car number realm location and car number recognition by image. At last the system will get the recognition result. If recognition believability is over 75%, then the system will conserve the image and recognition result, otherwise, it will conserve the car's all images and compress the recognition result, then pass those to the terminal system.

3. LICENSE REALM LOCATION AND CHARACTER INTELLIGENT SEGMENTATION

3.1 License realm location segmentation

The part of license location is a very important link in license recognition, In this paper we adopt text image fast-value method based on local extreme value, making fast two-value with import images, self-adapt form-filter, and extracting character target according to text's characteristics. For character target, we can adopt the following rational hypotheses, character size and strokes width are mostly the same; character interval is in the scope of permision; characters make up text-lines. So we can divide sub-realm with two-value image according to the value of every image data. The division regulations are: Image datum contained by every sub-realm have the same value; Image datum contained by any adjacent sub-realm have different value. Supposing the gather of target sub-realm O_i is Object, the gather of background sub-realm B_j is Background, and the gather of all sub-realm is:

$$\Omega = \text{Object} \cup \text{Background} = (\bigcup_{i=0}^{M} O_i) \cup (\bigcup_{j=0}^{M} B_j)$$

By character target's supposing, writing realm have veined grey feature which is different from background, and text realm has regular size. So by removing long-short-line noise whose vein don't conform to text target, we can correctly draw text realm. Suppose $L()$is the length of drawing element, the following is the steps of form-filter:

3.1.1 Removing non-character long-line sub-realm (level, vertical)

Choosing level size limit structure element Φ_h and vertical size structure element Φ_v ,making open operation with all elements in O_i , Supposing:

$$Object \Leftarrow \bigcup (O_i - O_i \cdot \Phi_h \cup O_i \cdot \Phi_v)$$

then get a new Ω through redividing 2-value images.

$$Background = \Omega - Object$$

3.1.2 Level filter

（1） Combing adjacent target sub-realm, choosing structure element Φ_{hi} ,then

$$L(\Phi_{hi}) = L_{hGap} + \lambda_{hm}(L(O_{i-1}) + L(O_i))$$

L_{hGap}: parameter of giving length(character interval estimate value),O_{i-1}， O_i : the adjacent realm of B_i, making open operation with B_i, and dividing the line again.

Background $\Leftarrow \bigcup_i B_i \cdot \Phi_{hi}$

(2) Combining adjacent background sub-realm, choosing level structure element Φ_{ht} ,then

$$L(\Phi'_{ht}) = L_{hTog} + \lambda_{hs}(L(B_i) + L(B_{i+1})) \quad (1)$$

L_{hTog}: parameter with given length (character strokes width With estimated value), B_i、B_{i+1}: The adjacent realm of O_i. making open operation with O_i, Dividing the line again

$$Object \Leftarrow \bigcup_i O_i \cdot \Phi'_{ht} \quad (2)$$

λ in formula (1) and (2) is parameter of target vein, which is used in self-adaptable regulation form of filter, L_{hGap} and L_{hTog} are parameters with given length, which are used to confirm the combining and noise-filtering structure element of form filter.

3.1.3 Vertical filter

The method of vertical filter is the same as level filter. By filtering long line, level filter, and vertical filter's operation, We will get target mark about the original image, from which we can easily use the method of projection analysis to locate the target realm.

3.2 Intelligent segmentation method instructed by recognition result

Usually, we can separate character with the method of fixed character intervals. By processing the isolated character with the method of line or row projection, we can get the outside rectangular of the isolated character, which can be the input of character classification module. But in real system, we may see some unusual situations such as dirty images, broken or overlapping characters. What's more, different car's categories and changeable colors and light changes caused by weather when testing, all of these increase the difficulty of license's segmentation and recognition. In this paper we adopt intelligent character segmentation method instructed by recognition result.

Definition 1: Function of character's width

$$\mu = (Width, CHAR\_W) = \begin{cases} e^{\frac{(width-CHAR\_W)}{2\sigma^2}}, & width \geq CHAR\_W \\ -e^{\frac{(width-CHAR\_W)}{2\sigma^2}} & width < CHAR\_W \end{cases}$$

In the formula, $\sigma = CHAR\_W / 3$

Definition 2: Recognition function of isolated character recog(x_0,y_0,x_1,y_1)=j, $j \in \bigcup\{M+1\}$, $\Omega = \{1,2,\cdots,M\}$ is a character category. j=M+1 indicates that the character is refused to be recognized, (x_0,y_0,x_1,y_1) is the coordinate of character realm.

Definition 3:Isolated character's recognition Bel(j)

$0 \leq Bel(j) \leq l$ indicate with recog (x_0,y_0,x_1,y_1) =j, and $j \in \Omega \bigcup \{M+1\}$,which determined by the method of recognition. To adherent characters, we introduce the following judging mechanisms of two possible separation dots: s_{k+1} and s_k.

$$if(\mu((S_{k+1} - S_k), CHAR\_W) > \theta_1 \ \&\& Bel(recog(x_0,y_0,x_1,y_1)) > \theta_2$$

then $C_K = (S_K, y_0, S_{k+1}, y_1)$

θ_1 can ensure the rational width of the separated characters, The judge of $Bel(\cdot) > \theta_2$ can ensure that the separated results are significant.

When the circumstance of broken characters appears, They can be judged by some shape parameters such as the ration of width and height, and depend on the system of recognition results to process the broken characters. To circumstances when adherent & broken characters both appear, We can combine the above 2 dividing & segmentation methods, that is, combination and segmentation at the same time, then, Satisfactory results will be reached.

4. CHARACTER RECOGNITION BASED ON NN

4.1 The establishment of Neural Networks' model

In the system of license real time recognition, because character images' background is very complicated, and some characters are dirty or incomplete, so the character samples we got contain a lot of noise. If we adopt ordinary Neural Networks recognition method, we may get unsatisfactory recognition structure. We establish networks' model which used in model match on the basis of Hamming Neural Networks' model put forward by Lippmann, which comprise of Matchnet and Recallnet (Figure 2). In study stage, Matchnet records and stores standardized sample models. In work stage, Matchnet will make a comparison between input models and sample models, and work out their similarities. By introducing the competition mechanism of "Central stimulation and Side suppress", we can select the best match models as classified results.

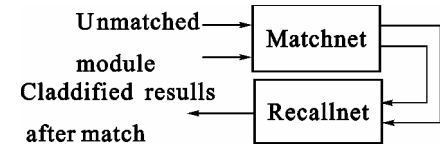


Figure 2 Improved Hamming Networks Structure

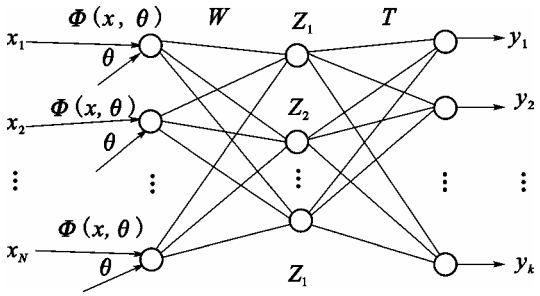


Figure 3 Improved Hamming Net Structure

Figure 3 is an improved Hamming Networks structure graph put forward in this paper. In the figure, the first layer's structure is the same to Hamming Networks, the second layer's structure is altered into Recallnet, which can restore the highest model categories in match scores.

4.2 Study of Networks

(1) According to the input samples, calculating

match score $Z_i = f(\sum_{j=1}^{n} W_{ij} X_j)$, $i = 1,2,\cdots,2^l$,

In this formula , the definition of function $f()$ is $f(x) = [(x+n)/n]^P$, P $\in$ N

(2) Calculating $y_i = g(\sum_{h=1}^{2^l} t_{ih} \cdot Z_h)$,

$i = 1,2,\cdots,K$ In this formula, the definition of function $g(\cdot)$ is $f(x) = \begin{cases} x & x > 0 \\ 0 & x \le 0 \end{cases}$

(3) Judging the input models to be what kind of sample model.

5. THE SCHEME OF CHARACTER RECOGNITION OF HIERARCHICAL MIXED INTEGRATOR

The paper adopts the method of character recognition comprising of several classifier for improving the system's recognition ration. The system comprises of English & Chinese character recognition Neural Networks, number recognition Neural Networks & a mixed Neural Networks. Character recognition Neural Networks are specialized in recognition number "1", English & Chinese characters which stand for provinces. We use the mixed Neural Networks to recognize some similar numbers and characters such as 5, 6, 8, etc. In order to improve the system's recognition rate and believability, we also need compare the output results with the database of car number.

6. CONCLUSION

We locate and cut number with the method of self-adaptable form filter, Making intelligent character segmentation instructed by recognition results, having established Neural Networks with Recallnet used in character recognition, and adopting some other methods, Such as the recognition system of hierarchical mixed integrator and rear-process correction, to improve the correction &ration of license recognition. Experiments show that the system has excellent property and stability.

7. REFERENCES

[1] Ye Xiangyun et. "Journal of Infrared and mm wave",1997,16(5):344.
[2] Zhao yang Lu. IEEE Trans, on Pattern Analysis and Machine Intell.,1998,32(10):4
[3] Zhao Xuechun, Qi Feihu. "Journal of Shanghai University of communication", 1998, 32(10).
[4] Bao Yuehan. "Neural Networks & model distinguished of self-adaptability", published by science, 1992.
[5] Hu Shouren., "Application of Neural Networks Technique", published by The National Defense Unversity Of Technology , 1993.

Teaching And Research Of Application Technology On Internet

Zhang Jianhua
School of Information & Science and Engineering,
Shenyang University of Technology,
Shenyang, 110021, P. R.China
E-mail: zhangjianhua2002@hotmail.com

ABSTRACT

This article illustrates the author's ideas and his understanding on the course setup, teaching plan and teaching methods of the Teaching and Research of Application Technology on INTERNET in the nation-wide universities of technology.

Keywords: Internet, research and teaching, application and technology.

1. BACKGROUND OF TEACHING AND RESEARCH OF THE COURSE ON INTERNET

With the dramatic development of application and technology on INTERNET, the process of economy information and socialism information is running rapidly. America is the most advanced country with the information technology and industry. How can we follow it quickly? First we must develop our economy, and the next is development of our education that is more important. Now, we must start the teaching and research on the methods of it, with concentrated attention.

In our country, the model of teaching must not be different, such as the courses setup and outline of teaching being decided, in the nation-wide universities of technology. Another problem is that the content of textbook is older compared the speed of technology development. Faced with this causes, some universities have brought causes of information technology into teaching and done a lot of beneficial work, such as opening optional courses. We think this way is the best and actually at present. The opened optional course is not only supplement of contents, and also is an important step, our students should get enough opportunity of going to society and taking part in the relative practical work..

Research on teaching of information technology is base on application technology on the INTERNET, majoring of Web and E- commerce, in the university now.

The concrete practice of this work has been for two years, and has been perfecting it step by step in our university. The advanced contents with flexibility and timely, are welcome by students. Sum up our experience, we think, the focal points is E-commerce but basis is HTML, we should keep attention to trends and choose the contents of next in the teaching course too. Actually the next is programming language C#, that conform to our situation now. We must pay attention to the method of teaching, bring every positive factor into play, foster the students' ability to study and work by themselves.

2. PLAN AND CONTENTS OF COURSES

That is serial courses, so we divide it three parts. First phase students must study base of knowledge. Second they must know well the base of application skill, and third they study advanced design way for applied in the INTERNET present now.

2.1 The phase of base knowledge studied

I all universities two level contents arranged in studying information technology, are "Computer Base Of Knowledge" and "Computer Base Of Application" according to our teaching plan, and students must complete them. That is prepared for next phase.

2.2 The phase of application skill studied

We must pay attention to the important phase and complete the plan with students together. The phase contain of three courses, "Design of Animated Drawing"、" Web Design Used HTML and DHTML " and "E- commerce and Programming Language". Consider of reduce class hours and simple the contents, we have merged the first and second to "Web Design and Programming Language" through practice. In this course, the content of "Design of Animated Drawing" is only explained simply as tool software.

2.2.1 Teaching Method of Web Design and Programming Language

HTML is the base of Web design. We are good at giving systematic guidance and finish it in 20 class hours. Next is CSS、Script and the base of Object-Oriented programming designing that is the concept of Object、Method and Event, finish in 10 class hours. It is preparing knowledge for next phase and necessary to study DHTML too. At last we explain XML and XHTML using two class hours, which is preparing knowledge for "E- commerce and Programming Language". In classes we recommend some outstanding Web designing works in the world to students, and auxiliary tool software such as Flash、GIF Animator etc.

If someone considers tool software such as FrontPage as content of "Web Design" in the phase, we don't think it's a good way in teaching. Although study well easily and use it quickly, deeply as base knowledge is very difficult. So open the lesson of E- commerce And Programming Language is very difficult too, because ASP and database application in INTERNET are all interrelated HTML. If studying HTML after finished FrontPage, it's worst as take the branch for the root, we think.

So open a serial course must be attention to one connected other and all contents are advanced for our final target.

2.2.2 Teaching Method of E-commerce And Programming Language

The contents of the course are about three sections: Design Personal Web Site、Commercial Web Site and Example For Application In INTERNET.

First section, students study how to setup Personal Site used tool software FrontPage, and finished it in 4 class hours. Another software for Personal Site used as self-study

materials.
Second section, they study how to setup Commercial Site from one of principles Client/Server system of INTERNET to ASP (Active Server Pages) programming in Server. The major is Object inlaid ASP and
ActiveX Component; next is application of database MS Server7.0 and at last is the technology of using ADO (ActiveX Data Objects) visit database in Web page.
Third section, there are 2 to 3 examples of application in INTERNET introduced is better for beginner and understanding program steps. If they want to be a skilled programmer, they must do a lot later.

2.3 The phase of studying advanced design method

Microsoft recommends the new software family NET Framework in 2000, and claim it will be suitable application for 10 years later. Last year, it is introduced to our country. It integrates all application programs using object model programmed different programming language, running different operating system. Actually it will be suitable server and application in INTERNET, and open up the family Windows Operation System too. The advanced design method must be learned and know well by our students.

2.3.1 The Contents of Courses

There are three software being suitable our university now which are Visual Basic.NET、Visual C++.NET and new programming language Visual C# .NET. The Object-Oriented programming language C# based on XML and increase ability of server in INTERNET. It absorbs the cream of C and C++; and has a simpler structure than C++. The students have learned Language C as the basic knowledge related to computer programming in the university, so studying C# is easier.

2.3.2 Teaching Method of C#

We can explain the base of C# and C++, and distinguish between them used six class hours first. Next studying Application Programming for open up Web and Operating Database used 14 to 16 class hours; Application Programming for open up Windows used 6 to 8 class hours. At last used 4 hours for the programming of Application in Multimedia.

3. LOOK INTO THE FEATURE OF APPLICATION TECHNOLOGY IN INTERNET

With the development of information technology and industrial, the new requirement will be produced. Web application in wireless translation industry has begun, and VRML applied in INTERNET will bring greater developing space for it. We must shoulder responsibility and go on improve the teaching syllabus and schedule in universities.

4. REFERENCES

[1] Zhang Zhiwen, “IE5 DHTML Design And Example”, People’s Post and Telecommunications Press, Sept. 1999
[2] Zhao Fengnian, “HTML CSS And DHTML”, Mechanization Industry Press, Aug. 2000
[3] “Introducing Microsoft .NET”, Microsoft Press, May 2001
[4] Liu Hao, Chen Sudong, “Technology And Example For C# Programming”, Tsinghua University Press, Feb. 2002
[5] John Shop, Jon Jagger, “Visual C#. NET Programming”, Peking University Press, Apr. 2002
[6] Zheng Jiajun, Chui Weining, Wang Danwei, “Visual C# and Programming with Examples”, Peking Hope Electronics Press, Aug. 2002.

The Uk E-Science Programme: Applications And Middleware

Tony Hey
Director UK e-Science Core Programme
EPSRC,
Polaris House,
North Star Avenue
Swindon SN2 1ET, UK
E-mail: tony.hey@epsrc.ac.uk
And/Or
Anne E. Trefethen
Deputy Director UK e-Science Core Programme
EPSRC,
Polaris House,
North Star Avenue
Swindon SN2 1ET, UK
E-mail: anne.trefethen@epsrc.ac.uk

Keywords:e-Science, Grid, Data Management

1. INTRODUCTION

e-Science refers to scientific collaboration which may involve scientists working together at distinct laboratories across the country, or even world, and may require access to distributed computing and data resources, remote access to specialized and expensive facilities and world-wide collaborations of scientists. The infrastructure to enable this science revolution is generally referred to as the Grid [1]. Whereas the Web provided easy access to distributed information through HTML pages; the Grid will provide seamless access to a much wider range of distributed resources and enable the formation of transient "Virtual Organisations" without compromising security or privacy.

In April 2001, the UK government, recognizing the potential of e-Science, started a new e-Science research programme. The programme had £120M available to support applications and middleware development. The majority of funding, £74M, was used to fund application orientated e-Science research, £9M was toward a new high performance computing facility and the remaining £35M was provided for generic research and support in a Core Programme.

Each of the five Research Councils - the Biotechnology and Biological Sciences Research Council (BBSRC), the Engineering and Physical Sciences Research Council (EPSRC), the Economic Social Research Council (ESRC), the Medical Research Council (MRC), the Natural Environment Research Council (NERC) and the Particle Physics and Astronomy Research Council (PPARC) - was given an allocation to fund applied scientific research, which would 'stretch' the Grid. The Core Programme was required to develop a computing and communications infrastructure to facilitate this scientific research

A further key component of the Core Programme remit was also to engage UK industry in the e-Science programme. The reasons for this were two fold: (1) It is essential that the research middleware development is in line with developments within industry, otherwise it will all be for nought; (2) It is important that industry is aware and ready for grid technologies as they become available to the broader community – this will ensure that UK Plc is not left behind in the take up of these new technologies.

In the following sections we will describe some of the scientific applications underway and the steps the Core Programme have taken to provide the required infrastructure.

2. E-SCIENCE APPLICATIONS

In this section we will review a number of the projects that have been funded in the e-Science programme. These should be considered as exemplars; space does not permit an exhaustive coverage of all projects.

A number of e-Science applications have been funded in the field of engineering. One such project is the DAME project [2] which considers the problem of health monitoring of industrial equipment. The consortium comprises a number of universities and industrial partners and the particular problem on which they are focused is the analysis of sensor data generated by Rolls Royce aero-engines – from pressure, temperature and vibration sensors. There are many thousands of Rolls Royce aircraft engines currently in use and each trans-Atlantic flight made by each engine generates on the order of a Gigabyte of data per engine. A small subset of this data is transmitted during the flight and the goal of the project is to capture, analyse and compare it with engine data stored in data centers around the world. It is hoped that the analysis of the data during the flight will identify early onset of mechanical problems. Rolls Royce hope to be able to lengthen the period between scheduled maintenance periods thus increasing profitability. The engine sensors will generate many Petabytes of data per year and decisions need to be taken in real-time as to how much data to analyse, how much to transmit for further analysis and how much to archive. The project combines issues regarding remote sensing, data management and real-time computational analysis.

The problem area in DAME could be thought of as health care monitoring for aircraft. Clearly there are applications of this sort in health care for society. The programme has funded a number of medical research and healthcare projects. CLEF [3] is an MRC funded project is concerned with developing a clinical framework for health informatics. Here again the issue is the capture, analysis and archiving of health related data, in this case patients health records. Naturally such a framework needs to be able to deal effectively with different types of data – digital patient records, clinical data, and images and needs to provide an effective interface to the clinicians, nurses and patients alike.

There are three EPSRC funded projects that are of particular

interest for bioinformatics and drug discovery. These are the myGrid [4], Comb-*e*-Chem [5] and DiscoveryNet [6] projects. These projects emphasize knowledge management, data federation, integration and workflow, and are in part building middleware services that will automatically annotate the experimental data as it is produced.

The Comb-*e*-Chem project is concerned with the synthesis of new compounds by combinatorial methods. The combinatorial approach entails synthesis and testing of new families of compounds and considering how they fit into the larger picture by comparison of previously understood compounds. Hence this project intends to develop an integrated platform that combines existing structure and property data sources within a grid-based information-and knowledge-sharing environment, together with the National Crystallographic Service at Southampton University, which can produce hundreds of samples per month The first requirement for this platform is to support new data collection, which includes information regarding the process used to create the data as well as the data itself. The next step is to integrate the experimental data, simulation modeling and data stored in databases, in an environment that provides a unified view of resources, with transparent access to data retrieval, online modeling, and design of experiments. Again we see in this project a combination of remote instrument integration, data collection, database federation and modeling. Without the Grid this would not be impossible to achieve but would certainly be impractical.

The DiscoveryNet pilot is focused on high throughput. It aims to design, develop and implement an advanced infrastructure to support real-time processing, interpretation, integration, visualization and mining of massive amounts of time critical data generated by high throughput devices. The devices and technology under consideration include biochips in biology, high throughput screening technology in biochemistry and combinatorial chemistry, high throughput sensors in energy and environmental science, remote sensing and geology. A number of application studies are included in the project - analysis of Protein Folding Chips and SNP Chips using LFII technology, protein-based fluorescent microarray data, air sensing data, renewable energy data, and geohazard prediction data. The project can be typified by large amounts of data requiring real analysis.

The myGrid project has a large consortium of five Universities, with the European Bioinformatics Insitute and industrial collaborators - GSK, AstraZeneca, IBM and SUN.

The goal is to develop an e-Scientist's workbench that will support: the scientific process of experimental investigation, evidence accumulation and result assimilation in the bio-sciences. A novel feature of the proposed workbench is provision for personalisation facilities relating to resource selection, data management and process enactment. Like other projects there are elements of workflow analysis, metadata and ontologies, and provenance and archiving. In this case the data under consideration are heterogeneous in that they are in many forms, text, numerical, images etc. myGrid will develop two application environments, one that supports the analysis of functional genomic data, and another that supports the annotation of a pattern database. The workbench will provide the environment in which the scientists area able to share effectively 'community' information and allow collaborations of virtual, dynamic groupings to tackle emergent research problems.

Astronomers around the world have very similar problems to those described above in the engineering and bioinformatics projects. At present, astronomical data using different wavelengths are taken with different telescopes and stored in a wide variety of formats. The amounts or data are large. A number of projects have been funded around the world, AstroGrid[7] in the UK, NSF NVO[8] in the USA and EU AVO [9] in Europe to try to tackle thses problems. The goal of these projects is to provide uniform access to a federated, distributed repository of astronomical data spanning all wavelengths from radio waves to X rays. They want to create something like a 'data warehouse' for astronomical data that will enable new types of studies to be performed. Again, the astronomers are building Grid infrastructure and services to support these Virtual Observatories.

It is clear in these applications and others that the new generation of hardware technology will generate data faster than humans can process it and it will be vital to develop software tools and middleware to support knowledge and information management, annotation and storage. Another example of this is the production of X-ray data by electron synchrotron accelerators. The present generation of accelerator creates an image every 3 seconds and hence each experimental station generates about 1 Terabyte of X-ray data per day. At the next generation 'DIAMOND' synchrotron currently under construction [10], the planned beamlines will generate many Petabytes of data per year, which will need to shipped, analysed and curated.

In order to ensure that the directions for middleware development were set correctly, the Core Programme created a Grid Architecture Task Force (ATF) [11] and Grid DataBase Task Force (DBTF) [12]. They were tasked with producing an 'e-Science Grid Road Map' for Grid middleware development. In analyzing the state of Grid middleware, it became clear at any early stage of the programme that it would be crucial to focus on the data issues [13]. There were few middleware services available to deal with federation and remote access of databases. As indicated above there is al need to go beyond the mechanisms of data federation and access to develop intelligent means of dealing with huge amounts of experimental data and effective use of resources on the Grid. We therefore have funded a number of projects to try to address these requirements. In the next section we will provide an overview of these projects, in the context of other middleware initiatives, and the building of a UK e-Science Grid.

3. D MIDDLEWARE and the UK e-SCIENCE GRID

In order to have the whole of the UK programme begin on an equal footing we provided projects with a common starting point for Grid middleware. The initial Grid middleware selected was that used by NASA in their Information Power Grid (IPG) [14] and includes Globus [15], SRB (Storage Resource Broker) [16] from San Diego, and Condor [17] from Wisconsin. The IPG connects the NASA laboratories across the USA. Since they are all part of one organisation they are able to support a homogeneous set of resources.

The UK e-Science Grid comprises a National e-Science Center located at Edinburgh and eight regional centers located across the UK. The centers are based at universities and hence the UK e-Science Grid connects resources at the university centers, which range from supercomputers, commodity clusters, databases and visualization services with different IT policies, firewalls and so on and hence stretches this basic infrastructure and the digital certificate based security system provided by Globus. Creating the UK e-Science Grid has meant the creation of a central certificate authority to provide digital certificates for the projects and center resources. The Certificate Authority has been set up as

part of a Grid Support Centre (GSC) that also operates a telephone and email help desk and software download site. The GSC lead an Engineering Task Force comprising a member from each of the centers, tasked with the engineering of the Grid between the centers. Building a grid between multiple sites that accommodate different resources, owned by different groups or individuals, each with their own policies, raises not only a number of technical issues but also many social barriers that need to be traversed.

Open source prototypes of the Grid middleware are available and under development as part of the e-Science programme and other international efforts. However, until recent months the standards being adopted and developed within the scientific grid community, at the Global Grid Forum [18], were quite separate from those developed within the World Wide Web Consortium and other bodies active in the web services area. Although the separate groups were indeed attempting to provide infrastructure for slightly different purposes, there was a clear overlap in the functionality they were developing. The potential divergence of technologies was a cause for concern, for the obvious reason that potential synergy between the two standardization processes was being lost. The Open Grid Services Architecture (OGSA) [19] essentially provides a merging of the two strands of activity. This grid services architecture will provide a unifying framework and set of services on which e-Science activities and resources can be integrated. This 'Service Oriented Architecture' sees the Web services standards being extended to include the more dynamic requirements of the Grid to create Grid services that provide the foundations for higher level applications. The e-Science community in the UK is actively involved in the specification and development of this architecture.

A key component of the architecture being addressed in the UK is the definition and implementation of services for data base access and integration. The Core Programme has funded a major project in this area, OGSA-DAI [20]. The project is collaboration between computer scientists at three e-Science centres NeSC, Newcastle and Manchester, and includes IBM and Oracle as industrial partners. A key to the success of the project is the provision of services that match the requirements of the pilot projects described above and hence there are several 'early adopters' who work with the development team in an iterative fashion employing and testing the implemented services. The myGrid and AstroGrid projects described above are both included in the early adopters for this project. The group involved in the project has initiated a new working group at the Global Grid Forum (GGF), OGSA-DAIS, where the initial specifications of the services have already been discussed.

The OGSA-DAI services provide the basic infrastructure for data manipulation and analysis, but in order for many e-Science projects to be successful, there is a need for the particular community to work together to define agreed XML schemas and other standards. The existence of such standards for metadata will be vital for the interoperability and federation of data held in different formats in file systems, databases or other archival systems. In order to construct 'intelligent' search engines, each separate community and discipline needs to come together to define generally accepted metadata standards for their community data grids. Since some disciplines already support a variety of existing different metadata standards, we need to develop tools that can search and reason across these different standards. It could be said that this would move the discussion beyond information to knowledge management and to the construction of genuine 'semantic grids' [21, 22]. The myGrid project is considering some of these issues, ontology development and the like, for the biosciences. We have funded a small project Molecular Informatics through the Cambridge e-Science centre [23] to develop these semantics for a part of the chemical community. The astrophysicists, AstroGrid, NVO, AVO, have agreed to work together to create common naming conventions for the physical quantities stored in astronomy catalogues. These agreed semantic tags will then be used to define equivalent catalogue entries across the multiple collections with the astronomy community. The medical, environment and other communities are all beginning to put in place such efforts.

These developments assist with the data that is already archived in databases and the like. With the imminent data deluge, as described above, the issue of how we handle this vast outpouring of scientific data becomes of paramount importance. To date experimental data has generally been manually managed and analysed to identify potentially interesting features and discover significant relationships between them. Given the amounts of data now involved, we need to automate the discovery process - from data to information to knowledge - as far as possible. At the lowest level, this requires automation of data management with provenance tags and archiving. At the next level we need to move towards automatic information management, which requires automatic annotation of scientific data with metadata that describes both interesting features of the data and of the process by which the data were formed. Finally, the step discussed above is to progress beyond structure information towards automated knowledge management of our scientific data. This will include the expression of relationships between information tags as well as information about the storage and organization of such relationships. Again many of the developments need to be made within the context of the scientific community and we see attempts being made at each of these levels within the projects described above. The issue of provenance annotation also comes down to the actual instruments and so it is important that the vendors developing those instruments are also engaged in the projects to ensure full understanding of the issues and requirements.

4. CONCLUSIONS

The UK e-Science Programme has allowed the parallel development of scientific applications and supporting Grid middleware. We believe that this has provided an excellent framework to drive the development of middleware.

The e-Science Programme as a whole has initiated a broad spectrum of first class applications that are both scientifically challenging and also demanding of the Grid infrastructure. The addition of a Core Programme has provided essential technical support to scientists, centralized certificate authorization, key middleware development and the involvement of industry. At the present time there are over 50 companies involved in the programme through collaborations in projects.

Within the UK we have focused on the data issues involved in Grids, from data generation and management up through information and knowledge management. We are building key components of middleware that will provide services for database access and integration and within specific domains developing the ontologies and services required for knowledge management.

We believe that the e-Science programme will not only lead to new methodologies for scientific investigation, but in time will provide the foundation for a new generation of computing infrastructure to a much broader community.

5. REFERENCES

[1] Foster, I. and Kesselman, C. (eds.). The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999.

[2] The DAME Project www.cs.york.ac.uk/DAME

[3] The CLEF Project http://www.cs.man.ac.uk/mig/projects/current/clef/

[4] The myGrid Project: http://mygrid.man.ac.uk

[5] The Comb-e-Chem project: http://www.combechem.org

[6] The DiscoveryNet project: http://www.discovery-on-the.net

[7] AstroGrid: http://www.astrogrid.ac.uk

[8] The NSF NVO: http://www.nvo.org

[9] The EU AVO: http://www.eso.org/avo

[10] The DIAMOND project: http://www.diamond.ac.uk

[11] The Architecture Task Force http://umbriel.dcs.gla.ac.uk/nesc/general/teams/atf.html

[12] The Database Task Force http://umbriel.dcs.gla.ac.uk/nesc/general/teams/dbtf.html

[13] Watson, P., Databases and the Grid. Technical Report CS-TR-755, University of Newcastle, 2001.

[14] NASA Information Power Grid: http://www.nas.nasa.gov/About/IPG/ipg.html

[15] Foster I. and Kesselman C., Globus: A Metacomputing Infrastructure Toolkit, International Journal of Supercomputer Applications, 11(2): 115-128, 1997.

[16] http://www.globus.org

[17] Storage Resource Broker: http://www.npaci.edu/DICE/SRB SRB

[18] M. Litzkow, M. Livny and M. Mutka, 'Condor – A Hunter of Idle Workstations', Proceedings of the 8th International Conference of Distributed Computing Systems, pages 104-111, June 1988.

[19] http://www.cs.wisc.edu/condor

[20] Foster, I., Kesselman, C., Nick, J. and Tuecke, S., The Physiology of the Grid: Open Grid Services Architecture for Distributed Systems Integration, to be presented at GGF4, Feb. 2002.

[21] The OGSA-DAI Project, http://www.nesc.ac.uk

[22] The Global Grid Forum, http://www.gridforum.org

[23] D. DeRoure, N. Jennings and N.Shadbolt, Towards a Semantic Grid, Concurrency & Computation (to be published) and in this collection.

[24] R.W. Moore, Knowledge-Based Grids, Proceeding of the 18th IEEE Symposium on Mass Storage Systems and Ninth Goddard Conference on Mass Storage Systems and Technologies, San Diego, April 2001.

[25] The Cambridge e-Science Centre, http://www.escience.cam.ac.uk

LBGK simulation of the laminar flow around a square cylinder in a channel and its visualization *

Nengchao Wang[1,2], Weibin Guo[1], Baochang Shi[2]
[1]School of Computer and Science and Technology, Huazhong University of Science and Technology
[2]Parallel computing Institute, Huazhong University of Science and Technology
Wuhan 430074, People's Republic of China
E-mail: ncwang@public.wh.hb.cn
And
Zhaoli Guo
National Key Laboratory of Coal Combustion, Huazhong University of Science and Technology
Wuhan 430074, People's Republic of China
E-mail: pcihust@wuhan.cngb.com

ABSTRACT

This paper presents a detailed numerical investigation to the laminar flow around a square cylinder mounted inside a two-dimensional channel with a blockage ratio of β=1/8, a Reynolds number range of 1～300 using a newly developed incompressible nonuniform lattice-BGK model. It is found that the vortex shedding behind the cylinder induces periodicity in the flow field. Details of the phenomenon are simulated through computer flow visualization. A number of quantities such as Strouhal number and drag, lift coefficients are calculated and thoroughly compared with other methods. Excellent agreement shows that the model gives accuracy results for complex flows.

Keywords: Lattice-BGK method, Square cylinder, Strouhal number, Drag coefficient, Lift coefficient.

1. INTRODUCTION

Over the preceding decade, with the advance of computer hardware and developments in sophisticated numerical algorithms, it has become easier to solve complex flows, albeit of limited size. To make satisfactory progress in this area often more efficient numerical methods are required. The lattice-Boltzmann (LB) method is such a new and highly efficient computational method that offers flexibility and outstanding amenability to parallelism when modeling complex fluid flows. Compared with other traditional computational fluid dynamics (CFD) methods, such as the finite difference schemes, the finite element, etc., the major advantage of the LB approach is that it provides insight into the underlying microscopic dynamics of the physical system investigated, whereas most methods focus only on the solution of the macroscopic equations [1,2]. In recent years, the LB method has been successfully used for simulating many fluid flow problems and for modeling physics in fluids [3-6].

The flow around bluff bodies in a two-dimensional channel has been an attraction in all kinds of fluid mechanical investigations for a long time. This flow situation is popular not only because its academic attractiveness but also owing to its related technical problems associated with the energy conservation and the structural design of buildings, bridges, towers, cars, masts and wires, etc.[7-11]. Much work has been done in simulating 2D flow around such bluff obstacles in past. In particular, 2D flow around circular cylinders has been studied extensively. In contrast to many theoretical, experimental, and numerical data on the flow around circular cylinders over a wide range of Reynolds numbers, there are very few similar studies and information on the flow around square bodies [8,11]. Previous investigation of the flow around circular cylinders performed with the LBM clearly shows that this method is an appropriate tool for such kinds of flows [12].

In this paper we investigate the vortex-shedding phenomena and the topology of the vortex structure behind the square cylinder in a two-dimensional duct change with the Reynolds numbers with a goal of providing reliable results from simulations. Again, the influence of the Reynolds number on the lift and drag forces is also one purpose of the present work. The simulating results for a range of Reynolds numbers between 1 and 300 are presented and thoroughly compared with other numerical data with respect to the Strouhal number, lift and drag coefficients.

2. THE PROBLEM

The 2D laminar flow around a square cylinder with diameter D mounted centered inside a plane channel (height H) was investigated (see Figure 1). The inflow is located X_u units upstream of the cylinder. The outflow is located X_d units downstream of the cylinder. The size of the obstacle, D, and the channel height, H, define the solid blockage of the confined flow (blockage ratio $\beta=D/H$).

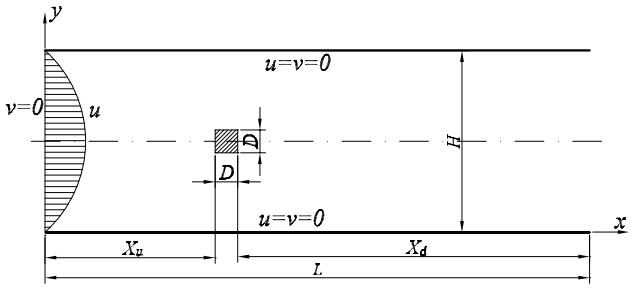


Figure 1 Computational domain and boundary conditions

The governing equations used in the simulation were the 2D non-dimensional Navier-Stokes equations

* This work is supported by the National Natural Science Foundation of China (Grant No.60073044) and the Special Found for Major State Basic Research Project in China (Grant No. G1999022207).

$$\nabla \cdot u = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + \nabla \cdot (uu) = -\nabla p + Re^{-1}\nabla^2 u \tag{2}$$

where $Re = u_{\max} D/\upsilon$ is the Reynolds number, $u_{\max}$ is the average flow velocity of the parabolic inflow profile and υ is the kinematic viscosity.

The boundary conditions in this investigation are as follows. The inlet velocity field was specified as a parallel flow with a parabolic horizontal component given by $u(y)=4y(1-y)$ for $1 \leqslant y \leqslant 1.0$. This produces a maximum inflow velocity of $u_{\max}=1.0$. The outflow boundary condition assumed a parallel flow and $\partial u/\partial x = \partial v/\partial x = 0$. At the top and bottom surfaces of the channel, $u=v=0$. No-slip boundary conditions were prescribed for all solid surfaces. The normal derivative for the pressure was set to zero at all boundaries. The normal derivative in the diagonal direction for the pressure was also set to zero at all corners of the flow field.

3. THE NUMERICAL METHOD

The lattice Boltzmann model used in the present simulations is the 9-bit incompressible lattice Bhatnagar-Gross-Krook model (i*d2q9*) proposed by Guo [12] is used. The directions of the discrete velocity used in the model are given by $e_0 = 0$, $e_i = \left(\cos[(i-1)\pi]\pi/, \sin[(i-1)\pi]\pi/\right)$ for i=1:4, and $e_i = \sqrt{2}\left(\cos[(i-5)\pi/2+\pi/4], \sin[(i-5)\pi/2+\pi/4]\right)$ for i=5:8. The evolution equation of the distribution function $g_i(x,t)$ reads

$$g_i(x + ce_i\Delta t, t + \Delta t) - g_i(x,t) = -\frac{1}{\tau}[g_i(x,t) - g_i^{(0)}(x,t)] \tag{3}$$

where $c = \Delta x/\Delta t$ is the particle speed, Δx and Δt are the lattice spacing and the time step, respectively. τ is the dimensionless relaxation time. $g_i^{(0)}(x,t)$ is the equilibrium distribution function defined by

$$g_i^{(0)} = \lambda_i p + \omega_i[3\frac{(e_i \cdot u)}{c} + 4.5\frac{(e_i \cdot u)^2}{c^2} - 1.5\frac{|u|^2}{c^2}], \tag{4}$$

where $\omega_0 = 4/9$, $\omega_i = 1/9$ for i=1:4, and $\omega_i = 1/36$ for i=5:8. $\lambda_0 = -4\sigma/c^2$, $\lambda_i = \lambda/c^2$ for i=1:4, and $\lambda_i = \gamma/c^2$ for i=5:8. σ, λ, and γ are parameters satisfying $\lambda + \gamma = \sigma, \lambda + 2\gamma = 1/2$.

The kinematic viscosity is determined by $\upsilon = \frac{(2\tau-1)}{6}\frac{(\Delta x)^2}{\Delta t}$. The macroscopic flow velocity and pressure are given by

$$u = \sum_{i=1}^{8} ce_i g_i \,,\; p = \frac{c^2}{4\sigma}[\sum_{i=1}^{8} g_i + s_0(u)] \cdot \tag{5}$$

In order to capture some small scale phenomena in the simulations, we used a newly developed nonuniform lattice-BGK model, DDLBM [13], based on the Domain Decomposition Technique. The basic idea of DDLBM is to decompose the whole flow field into some relative regular subdomains, and each subdomain is then covered with a uniform grid on which a uniform lattice model is based. The interpolation between the interfaces is carried out in order to couple the subdomains consistently. The DDLBM solves flow problem by general lattice-BGK model in each subdomain and takes full advantage of the uniform LB models. From a computational point of view nonuniform grids can be efficient for computing fluid flows because the grid resolution can be adapted to the spatial complexity of the flow dynamics. An additional advantage of the domain-decomposition LB model is the good coarse-grain parallelism.

4. VISUALIZATION OF FLUID FLOW

The numerical simulations of the complex flow often generate massive amounts of data that must be interpreted for accurate prediction and understanding. Flow visualization is an important tool to assist the analysis of the simulation results. In the present work, a flow visualization system was implemented according to the Object-Oriented programming paradigm, allowing the developer to create his own representation of the simulated flow as contour images of scalars or vector plots.

In this paper we incorporate Visualization Toolkit (VTK) [14] into our application as a visualization engine. The VTK is an open source, freely available software system for 3D computer graphics, image processing, and visualization. The graphics model in VTK is at a higher level of abstraction than rendering libraries like OpenGL or PEX. This means it is much easier to create useful graphics and visualization applications. It supports a wide variety of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods; and advanced modeling techniques like implicit modeling, polygon reduction, cutting, contouring, etc. In VTK applications can be written directly in C++, Tcl, Java, or Python.

5. RESULTS AND DISCUSSIONS

The numerical simulations were performed on a 80×500 lattice for a Reynolds number range $1 \leqslant Re \leqslant 300$ and a fixed blockage ratio β=1/8. In this Reynolds number range, it is known from experiments and other numerical studies that the flow past square cylinders can be considered as 2D. At a Reynolds number *Re* above approximately 300, the flow might become three-dimensional, and two-dimensional computations will therefore produce unphysical results [9]. Therefore, this Reynolds number was chosen as the upper limit of the present 2D laminar simulations. In simulations, all the results were normalized to allow comparisons between the present work and other results, velocities with $u_{\max}$, physical times with $D/u_{\max}$ and frequencies with $u_{\max}/D$. For all cases considered, obstacle of size being 10×10 lattice units was positioned vertically centered in the first third section of the computational domain. The computational region Ω was decomposed into two subdomains Ω_1 and Ω_2 (see Figure 2). Here, Ω_1 was a square domain whose left-side and right-side virtual boundaries were *BL*=156 and *BR*=196, respectively. The lower and upper boundaries were *BD*=20 and *BU*=60, respectively. The subdomain Ω_2 was $\Omega_2=\Omega-\Omega_1$. The lattice size for Ω_1 was 160×160. The i*d2q9* model was used on Ω_1 and Ω_2. Hence, the refinement factor was $n=\Delta x_2/\Delta x_1 = 4$.

The Reynolds number and the relaxation parameter were used as input parameters. The relaxation parameter $w=1/\tau$ for *Re*=1,40,62,100,200,300 was set to be 0.3,1.0,1.2,1.4,1.7, and 1.8, respectively. The pressure was initialized to be p=0, and the velocities at all nodes, except for the nodes at the inflow boundary, were set as $u=v=0$.

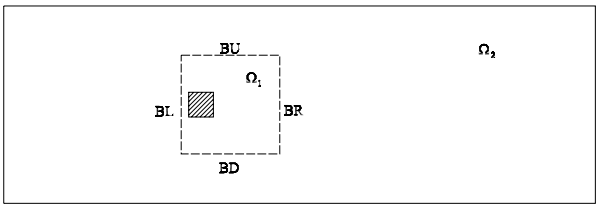


Figure 2 Domain decomposition

The flow with Re=0.5 was first simulated. The streamlines contours are shown in Figure 3. Figure 4 shows the vorticity contours corresponding to the streamline patterns presented in Figure 3. As shown in Figures 3(a) and 4(a), for $Re \leqslant 1$, the creeping steady flow past the square cylinder and no separation takes place. As Re increases until a certain value, the flow separates first at the trailing edge of the cylinder and a closed steady recirculation region is observed. The length of the recirculation region increases linearly with Re (Figure 5). The flow pattern is perfectly symmetric with respect to the oncoming flow and vortex shedding has not started. Whereas for $Re>Re_c$, where Re_c is between Re=60 and 62, the symmetry eventually breaks up. With increasing Reynolds number, after a sufficient number of iterations, the flow becomes periodic and evolves with a fixed frequency f. The well-known von Kármán vortex street with periodic vortex shedding from the cylinder can be detected in the wake. At Re=80, the separation point of the vortices is observed to be the rear edge of the obstacle and moves from the rear to the front edge of the obstacle with higher Reynolds numbers. At Re=133, small secondary vortices can be found at the top and bottom of the obstacle and the separation is observed at the leading edge of the square cylinder. The flow field in front of the cylinder for Re=80-300 is nearly independent of the structure of the wake. Within this range of Reynolds numbers, the above plots also show clearly the effect of the Reynolds number on the flow pattern. All the flow structures in our simulation compare well with the previous flow visualization and numerical simulations.

To quantify the results, the Strouhal number St is plotted as a function of the Reynolds number Re in Figure 6. The Strouhal number is defined as $St= f_sD/u_{\max}$, where f_s is the vortex shedding frequency and determined by spectral analysis of the temporal evolution of the velocity component v at several points in the wake behind the obstacle or measured from the time evolution plot of the lift coefficient C_l. Figure 7 depicts the time evolution of the lift coefficient at Re=100. As seen from Figure 6, the present method yields a somewhat lower Strouhal number for the entire range of Reynolds number compared with, but the present Strouhal number distribution fits quite well with Ref. 8, 11 and 15. Our method, furthermore, shows that the Strouhal number has no dependence on grid resolution.

The drag coefficient C_d is shown in Figure 8. Figure 9 compares the variations of the drag coefficient ($\max(C_d)$-$\min(C_d)$) and the lift coefficient ($\max(C_l)$-$\min(C_l)$) for the Re range $1 \leqslant Re \leqslant 300$, respectively. As shown in figures, the present work provides reliable and accurate results. The drag coefficient, C_d, and the lift coefficient, C_l, are calculated using

$$C_d = F_x \Big/ \frac{1}{2}\rho u_{\max}^2 D \quad \text{and} \quad C_l = F_y \Big/ \frac{1}{2}\rho u_{\max}^2 D \,, \quad (6)$$

where F_x and F_y are the drag force and lift force on the cylinder.

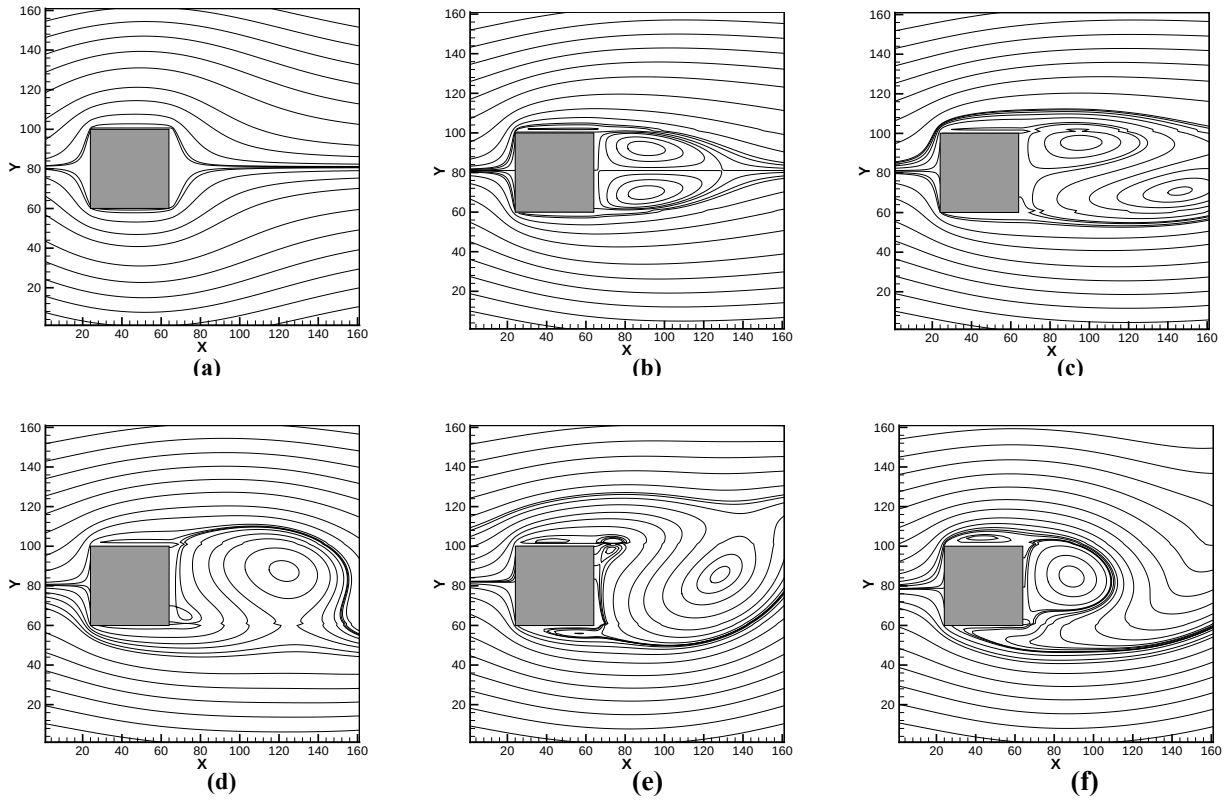


Figure 3 Streamlines around the square cylinder for different Reynolds numbers: (a)*Re*=1; (b)*Re*=40; (c)*Re*=62; (d)*Re*=100; (e)*Re*=200;(f)*Re*=300

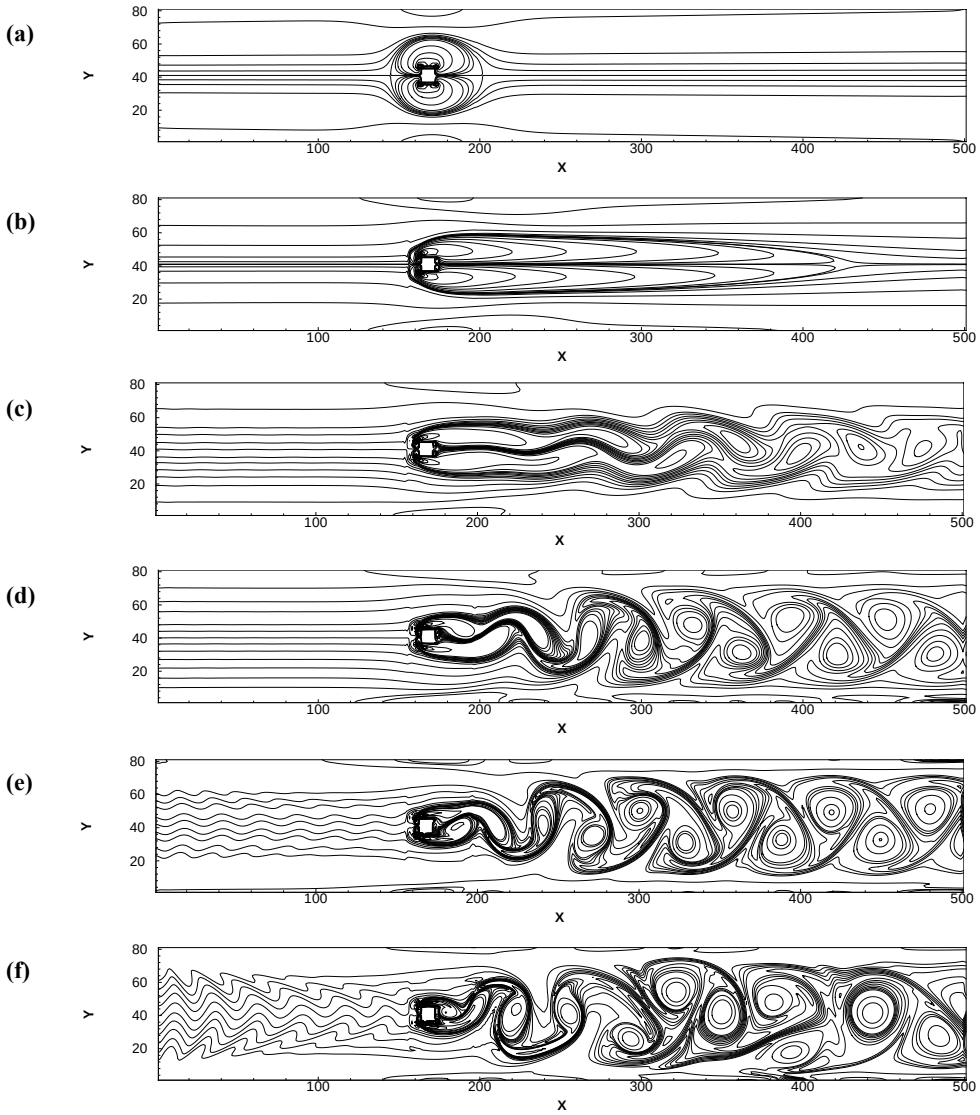


Figure 4 Vorticity contours around the square cylinder for different Reynolds numbers: (a)*Re*=1; (b)*Re*=40; (c) *Re*=62; (d)*Re*=100; (e)*Re*=200;(f)*Re*=300

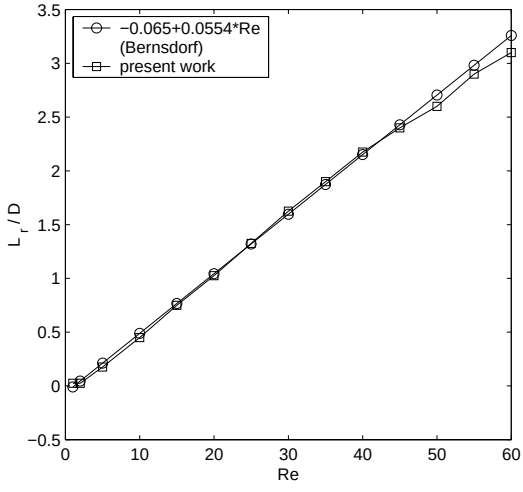


Figure 5 Recirculation length L_r vs. Reynolds number

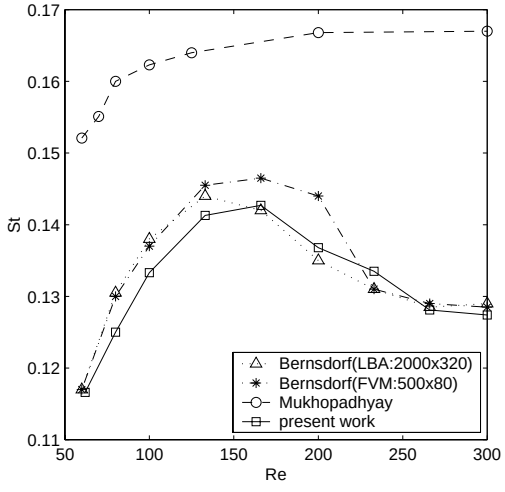


Figure 6 Strouhal numbers *St* vs. Reynolds number

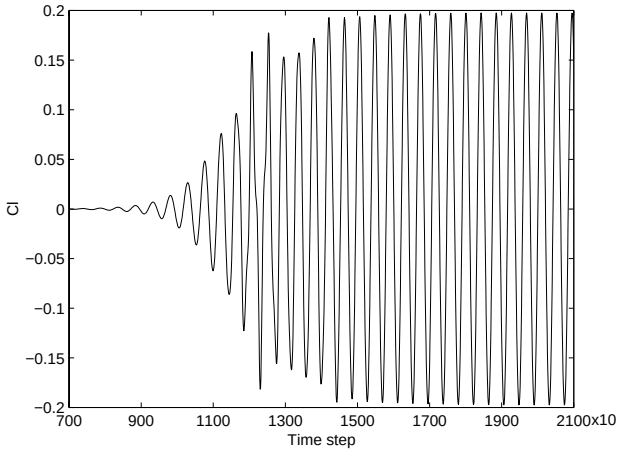


Figure 7 Time evolution of lift coefficient

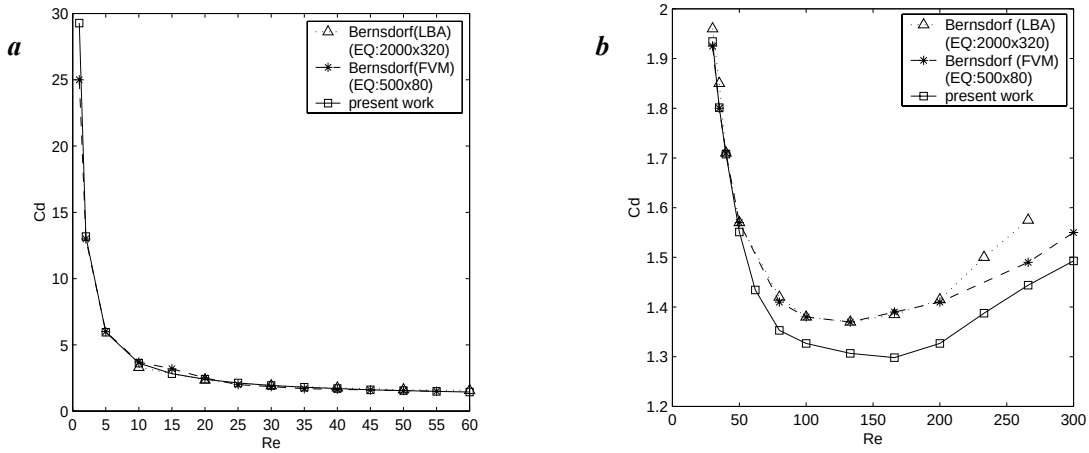


Figure 8 Computed time-averaged drag coefficient C_d vs. Reynolds number Re: (a) steady flow; (b) unsteady flow

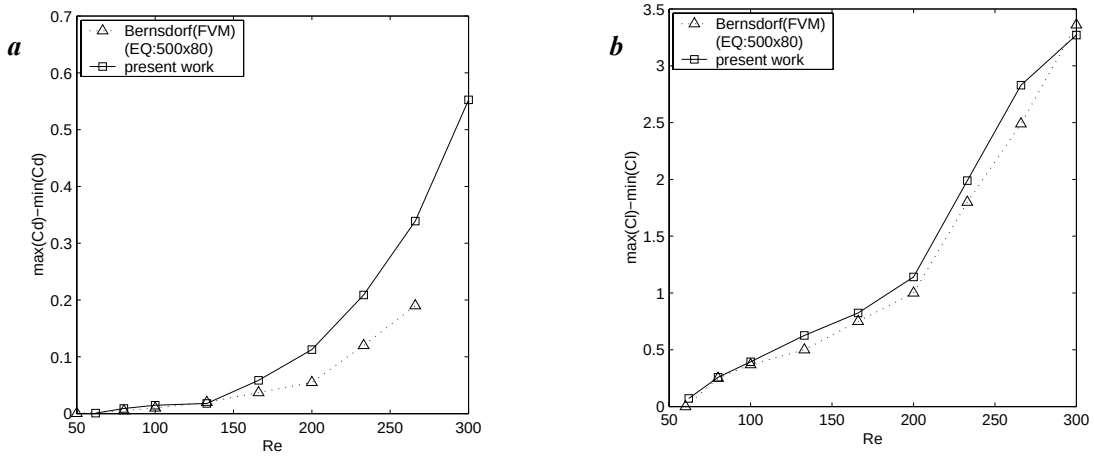


Figure 9 Variation of force coefficient vs. Reynolds number: (a) drag variation, max(C_d)- min(C_d); (b) lift variation, max(C_l)- min(C_l)

6. CONCLUSIONS

Numerical simulations of incompressible 2D flow around a square cylinder inside a channel (β=1/8) have been carried out in the Reynolds number range $1 \leqslant Re \leqslant 300$. In order to generate reliable numerical results, a newly developed incompressible nonuniform lattice-BGK model was applied. The integral parameters such as recirculation length, Strouhal numbers, drag and lift coefficients were computed and compared with the scattered data in the literature. We were able to show that our implementation of the lattice-BGK approach yields reliable and accurate results. Compared with the uniform lattice-BGK model, the nonuniform lattice-BGK model reduced significantly the

computational time and can
be used to find some local details in the flow fields.
During simulation, since the maximum velocity and the lattice size are limited, the relaxation parameter, w, needs to be large to achieve the higher Reynolds numbers. It is found that the lowest value of w leading to stable simulations depends on the Mach number. The appropriate value of c is about 6.5～12. On the other hand, to obtain a reliable simulation, w should not be too close to its upper limit. It must less than 2 to ensure positive viscosity.
The simulations described in this paper have been performed on a distributed shared memory parallel computer using a standard PVM library. The code has been written in ANSI-C. Data parallelism turns to be a well suited programming model to implement the computational fluid flow simulations.

7. ACKNOWLEDGEMENTS

We would like to thank Prof. Q. P. Guo for his many helpful discussions and valuable suggestions.

8. REFERENCES

[1] B.C.Shi, N.C.Wang, W.B.Guo et al.. “LBGK Simulations of Driven Cavity Flow at High Reynolds Numbers”, Proc. of DCABES2001, Wuhan: Hubei Sci. & Tech. Press, 2001.
[2] B.C.Shi, Z.L.Guo, N.C.Wang. “LBGK Simulations of Turbulent Natural Convection in a Cavity”, Chinese Physics Letters, Vol.19, No.4, 2002, pp.515~519.
[3] B.Chopard, A.Masselot. “Cellular automata and lattice-Boltzmann methods: a new approach to computational fluid dynamics and particle transport”, Future Generation Computer Systems, Vol.16, 1999, pp.249~257.
[4] X.Y.He, L.-S.Luo. “Lattice Boltzmann model for the incompressible Navier-Stokes equation”, Journal of Statistical Physics, Vol.88, 1997, pp.927~944.
[5] Z.Lin, H.P.Fang, R.B.Tao. “Improved lattice Boltzmann model for incompressible two-dimensional steady flows”, Physical Review E, Vol.54, 1996, pp.6323~6330.
[6] Q.S.Zou, S.L.Hou, S.Y.Chen et al. “An improved incompressible lattice Boltzmann model for time-independent flows”, Journal of Statistical Physics, Vol.81, 1995, pp.35~48.
[7] X.Y.He, G.Doolen. “Lattice Boltzmann on curvilinear coordinates systems: flow around a circular cylinder”, Journal of Computational Physics, Vol.134, 1997, pp.306~315.
[8] J.Bernsdorf, TH.Zeiser, G.Brenner et al. “Simulation of a 2D channel flow around a square obstacle with lattice-Boltzmann (BGK) automata”, International Journal of Modern Physics C, Vol. 9, No.8, 1998, pp.1129~1141.
[9] A.Sohankar, C.Norberg, L.Davidson. “Low-Reynolds Number flow around a square cylinder at incidence: study of blockage, onset of vortex shedding and outlet boundary condition”, International Journal for Numerical Methods in Fluids, Vol.26, 1998, pp.39-56.
[10] J.G.Wissink. “DNS of 2D Turbulent flow around a square cylinder”, International Journal for Numerical Methods in Fluids, Vol.25, 1997, pp.51~62.
[11] A.Mukhopadhyay, G.Biswas, T.Sundarajan. “Numerical investigation of confined wakes behind a square cylinder in a channel”, International Journal for Numerical Methods in Fluids, Vol.14, 1992, pp.1473~1484.
[12] Z.L.Guo, B.C.Shi, N.C.Wang. “Lattice BGK model for incompressible Navier-Stokes equation”, Journal of Computational Physics, Vol.165, 2000, pp.288~306.
[13] Z.L.Guo, B.C.Shi, N.C.Wang. “A Nonuniform lattice Boltzmann method based on domain decomposition”, Chinese Journal of Computational Physics, Vol.18, No.2, 2001, pp.181~184
[14] W.Schroeder, K.Martin and W.Lorensen. The visualization Toolkit: An object-Oriented Approach to 3D Graphics, 2nd ed., New York: Prentice-Hall, Old Tappan, 1998.
[15] M.Breuer, I.Bernsdorf, T.Zeiser, F.Durst. “Accurate computations of the laminar flow past a square cylinder based on two different methods: lattice-Boltzmann and finite-volume”, International Journal of Heat and Fluid Flow, Vol.21, 2000, pp.186~196.

Development of Supercomputing Environment *

Hong Wu, Sun-gen Deng, Hai-li Xiao, Bo Chen, Xue-bin Chi
Supercomputing Center of Computer Network Information Center
Chinese Academy of Sciences, Beijing 100080, P.R. China
E-mail: {wh, dsg, haili, chb, chi}@jupiter.cnc.ac.cn

ABSTRACT

Accompanying with the scientific computation becoming more and more important, and having become the third means for scientific research successive to the theoretical and experimental methods, more and more powerful computational ability and user-friendly computing environment are needed. In recent years, the Grid technology has been developed rapidly, and a lot of interdiscipline, such as Commodity Grid Technology, Grid portal etc., have been put forward to producing all outcomes which Grid technology combines with other technologies. So the toolkits of Globus, Commodity Grid technology and GPDK are introduced briefly in this article. Combining the status of our center and our own needing, the architecture and function of our computational portal are proposed. The preliminary achievement in creating our own computational grid is introduced in detail. Finally, some conclusion and experience in the Computational Portal building are set forth.

Keywords: Superomputing environment, Grid technology, Globus toolkits, Commodity Grid technology, GPDK, Portal Building.

1. INTRODUCTION

For the developments of science and technology, the computation plays very important roles. Traditionally, the theoretical and experiment methods are major means for scientific research. As the effect of computation in science and engineering becomes greater and greater, the scientific computation has been regarded as the 3rd means for scientific research. The reason we need computation in scientific research is that there are many problems without the analytic solution. For example, if we want to get the value of π, the only way is to calculate an approximate value with any precision. As for scientific computation, the most important property is that a lot of computations have to be done, and in solving some actual problems, the computation always needs to be finished in a given and relatively short period. The requirements of computation make the computer to be faster and faster. For the past 3 decades, the parallel computer had developed quickly. Now the fastest parallel computer has reached 40 TFLOPS.

While the computing power becomes more and more powerful, the scale of problem to be solved also becomes larger and larger, the amount of data to be processed becomes unimaginable large. Present, the scientists do not only need a very fast computer or a supercomputer, but also need a user friendly computing environment, so in order to make full use of the expensive and scattered resources (hardware & software) and to obtain more powerful computational capability, it is necessary to develop some new computational environment, new computational methods and make the computer more powerful for the endless requirement. For this purpose, there are many concepts which are related to computing environments created in the literatures, such as metacomputing [1], Grid [2], NC [3], etc..

In recent years, the Grid technology has been developed rapidly, and current computing environment based on the grid technology is becoming popular. This kind of computing environments always focus on a particular subject, such as Bioinformatics grid, grid physics network and so forth. It is considered as that the grid technology will bring a big impact on information technology in the 21st century. It is also considered as the 3rd wave of information technology.

2. BACKGROUND

2.1. Grid Technologies & Globus

2.1.1 Grid problem, Grid technologies and its application

The Grid problem is defined as flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources (virtual organizations), and the unique authentication, authorization, resource access, resource discovery, and other challenges which encountered in such setting is addressed as Grid technologies [4]. Grid applications are distinguished from traditional client-server applications by their simultaneous use of large numbers of resources, dynamic resource requirements, use of resources from multiple administrative domains, complex communication structures, and stringent performance requirements, among others [5]. Over the past ten years or so, a great progresses have been made within the Grid community, such as new protocols, services, tools, etc.. Among the three classification of Gird: computational Grid, data Grid and service Grid, the computational Grid had been researched rather ripe compared with the other two.

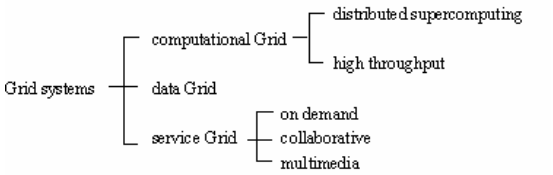


Figure 1 A Grid system taxonomy

2.1.2 Globus Project and its toolkits

To simplify the Grid application development task, the Globus Toolkit has been developed by providing implementations of various core services. The Globus Project is supported by The Defense Advanced Research Projects Agency (DARPA), The National Science Foundation (NSF), The National Aeronautics and Space Administration (NASA), which is started at 1996 and developed by Argonne National Laboratory's mathematics

* This work is supported by The Special Funds for Major State Basic Research Projects of China (G1999032805), State Hi-Tech Research and Development Program of China (2001AA111043).

and Computer Science Division and the University of Southern California's Information Sciences Institute. Its development is focused on low-level mechanisms that can be used to (a) implement higher-level services and (b) techniques that allow those services to observe and guide the operation of these mechanisms [6]. The achievement which have been obtained includes: development of Globus Toolkits, standardization of Grid technology, building of relative organization (such as, Work Group and Mailing List), a lot of Working Draft had been made also. A lot of Testbeds, which provide evaluating environment of function and performance of Globus Toolkit and support more advanced scientific research have also been built over the past 10 years. The long-term goal in the Globus project is to address the problems of configuration and performance optimization in metacomputing environments [7].

The Globus toolkit is a collection of software components designed to support the development of application for high-performance distributed computing environment, or "computational grid" [2,7]. And now the Globus Toolkit has become the accepted standard for developing the Grid-based application. The latest version of it is Globus Toolkit 2.0 (GT2.0), which was released in February 2002, and GT 3.0 public alpha release will be put forward at the end of 2002 [8]. The core services which compromise the toolkit can be pictured briefly in Table1.

Table 1 Core services in Globus Toolkit 2.0

Service	Name	Description
Resource Management	GRAM	Resource management protocol
Information Services	MDS	Metacomputing directory services/Monitoring and Discovery Services
Date Management	GridFTP	Data transfer protocol
Grid Security Infrastructure	GSI	Security protocol on all connections

2.2. Commodity Distributed Computing

Commodity technologies tend to focus on issues of scalability, component composition, and desktop presentation. The Commodity technologies and Grid technology are two parallel evolving directions. The result of combination of these two parallel evolutions is the occurring of Commodity distributed-computing technologies, which enable the rapid construction of sophisticated C/S applications. So the main goal of Commodity Grid (CoG) project are mapped out into enabling developers of Grid applications to exploit commodity technologies wherever possible and exporting Grid technologies to commodity computing. The first achievement of CoG project is the design and development of Commodity Grid Toolkit (CoG Kit), whose definition is in [9]: defines and implements a set of general components that map Grid functionality into a commodity environment/framework. The benefit of the CoG Kit is that it enables application developers to exploit advanced Grid services (resource management, security, resource discovery) while developing higher-level components in terms of the familiar and powerful application development frameworks provided by commodity technologies.

We use the latest version of Java CoG (0.9.13) in the building of our Computing Grid, but there is some incompatibility with Myproxy, which provides a server with client-side utilities to store and retrieve delegated X.509 credentials via the Grid Security Infrastructure (GSI). So we use the latest Java CoG off CVS.

2.3 Portal Development

Although we can realize the atomic Grid operations through the Globus Toolkits, the methods is still through the way of command line which specialized software need downloading, installing and configuring. Developer is seeking for a Problem Solving Environment (PSE)[10] to provide a personalized "home site" and make it more convenient to use for end-users, and web browser, the entry/starting point for WWW is adopted. As the Commodity Grid Toolkit was the result of combination of Commodity technology and Grid technology discussed above, the overlap and mixture of Grid technology and Web portal bring the concept of Gird portal. A Grid portal is defined to be a web based application server enhanced with the necessary software to communicate to Grid services and resources, which provides application scientists a customized view of software and hardware resources from a web browser[11].

There are mainly two middleware to develop Grid portal, one is Grid Portal Development Portal (GPDK [11]) and another is Gridport [12]. GPDK is implemented in Java and makes use of Java CoG for access to Grid services, and is developed by the Lawrence Berkeley National Laboratory, University of California, while Gridport is implemented in Perl and makes use of the existing HotPage technology for access to Grid services, and is developed by the San Diego Supercomputer Center (SDSC).

3. DEVELOP SUPERCOMPUTING ENVIRONMENT

There are 3 high performance computers in our center. They are SGI Power Challenge/16 processors, Hitachi SR2201/32 processors, and Dawning-2000/164 processors, respectively. Prior to the development of web-based supercomputing environment, users have to log in to each HPCs in turn and recall different commands to view resources and submit jobs. In order to provide a commodity distributed-computing environment and make full use of them, it is necessary to provide a customized, web-based view of software and hardware resources and convenient computing environment to the end-users.

3.1 The Architecture and Function

The first step of our work is to build a user-specific portal. To build an application-specific portal, which focuses on the high-performance computing, is our final goal.

As a whole, the function of our computational grid can be divided into two parts: user computing module and administrator management module, Taking consideration of the security and technology realization, we provide fairly simple function in the module of administrator management [13-14].

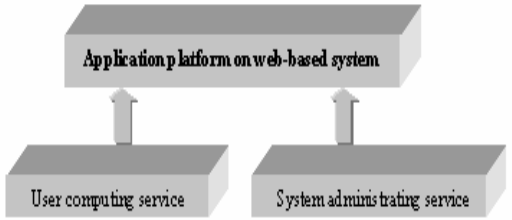


Figure 2 The main module of our Computational Grid

The function of administrator service includes: user management, status monitoring, log management, accounts management and other common operation, and the function of end-user service includes: file service, job service, resource view, accounts services, and also in order to enable large-scale scientific projects to better utilize distributed, heterogeneous

resources to solve a particular problem or set of problems, rich application software interface will be provided in the future.

3.2 Preliminary Achievement We obtained

3.2.1. Testbed Building

A four-nodes (each with 2 processors) homogenous (Redhat 7.3) cluster was built to simulate the pseudo-MPP machine, a DES-1016 10/100M switch is used to connect nodes all, the configuration of hardware and software was set out in Table 2.

Table 2 Hardware configurations of Testbed

Node number	Node 1	Node 2	Node 3	Node 4
Main-board	Supermicro 3700 CE	Superm icro 370 DL3	Superm icro 370 DL3	Superm icro 370 DL3
CPU	Pen-tium 3(1G)	Pen-tium 3(1G)	Pen-tium 3(1G)	Pen-tium 3(1G)
Memory	256M	256M	256M	256M
Hard disk	IBM 40G	SCSI 36G	SCSI 36G	SCSI 36G
IP address	159.226.*.* /192.168.0. 1	192.16 8.0.2	192.16 8.0.3	192.16 8.0.4
OS	Redhat7.3	Redhat 7.3	Redhat 7.3	Redhat 7.3
Software	OpenPBS_ 2.3.2(server /mom), mpich-1.2.4 ,GT2.0, Myproxy0. 4.4, jarkata-tom cat3.2.4, CoG, GPDK	OpenP BS_2.3. 2(mom)	OpenP BS_2.3. 2(mom)	OpenP BS_2.3. 2(mom)

3.2.2. Preliminary Achievement

As the GPDK can facilitate the development of Grid portals and provides several key reusable components for accessing Grid services, so we choose GPDK as our beginning of portal development. Now we have realized the following function on our testbed, to end-user: user login, job submission with interactive and batch mode (through globus-gatekeeper & GSISSH), job outputs view, some simple user profile view, and some simple information services view (Fig.3-4), to administrator: portal configure, resources updating and log view (Fig.5).

Figure 3 The login page of our Computational Grid

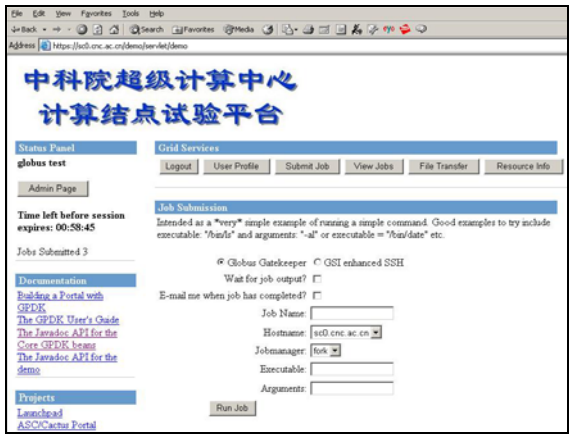


Figure 4 The job submission page of our Computational Grid

Figure 5 The administrator log-view page of our Computational Grid

3.3. Problem need to be overcome

The following are some technical problems during the development of supercomputing environment:

- Although most function can be provided in our portal now, but it is just limited into our Linux cluster, the next step is to install the Globus toolkits on each HPCs of our center, but the OS Version installed on our HPCs, which are IRIX 6.2, AIX 4.3.2…, respectively, are too old to be supported by Globus Toolkits. This problem is a great obstructer in building our own Computational Portal.
- MPICH-G2[15], the extension of the Argonne MPICH implementation of MPI to use services provided by the Globus Toolkits, is a grid-enabled implementation of the MPI. But most of our MPI are vendor MPI except the test cluster, so how to unify the MPI environment to make it grid-enabled is also a great problem.
- GridFTP, the Data Management pillar of Globus Toolkits, just provides a client tool called globus-url-copy at command prompt[16] now. There is not platform-specific client toolkits now is also a great restriction to us.

4. CONCLUSION AND EXPERIENCE

Operation of computing environment has experienced the console to telnet, till now via web-browser, all these changes just have one aim, which is to make the computing environment become more and more convenient to end-users.

The most distinct character of Grid technology which differentiates with former technologies and have close relationship with computing environment is the large-scale resource-sharing, that is also the most concerned problem in the field of distributed/super computing.

So we can draw the conclusion that the Grid technology has bright future from the above, and some of very good middleware Globus Toolkits, CoG Kits and GPDK are also developed which can be utilized to simplify our development task. And there are still some conclusions about technology in former development:

- Although the Globus Toolkits is a very good middleware, and has become the de facto standard in Grid building, but it is still lack of necessary stability and durative, this can be seen from the changing of sets of entries (which used in information description) in its fundamental information services, Globus Monitoring and Discovery Service (MDS), this change is especially reflected form MDS1.x to MDS2.x. And also the location changing of very important configuration files from GT1.x to GT2.0, till the latest version GT2.2 (which was released in Oct. 2002) is also a proof. So we can draw the conclusion of that the GT is now just a product of research project, which is far away from actual application.
- The phenomenon of incompatibility of necessary packages used in building a computational grid, such as GT, Myproxy, Java CoG, …, is rather great, that is to say, the necessary toolkit to build portal can not be developed synchronously, and that bring great difficulties to second-phase developers.
- GPDK, the other middleware, which we used to build our Computational Portal, is incompatible with the MDS2.1, (the important information service in GT2.0). There is still some problem in GPDK, such as its updating is too slow to develop synchronously with Globus Toolkits and CoG Kits, it's structure is not very clear, etc.. So it just can be seen as a rapid prototype tool.

5. REFERENCES

[1] C. Catlett and L. Smarr, Metacomputing, Communication of the ACM, Vol.35, No.6, 1992, pp.44~52.

[2] I. Foster and C. Kesselman, editors. The Grid: Blueprint for a Future Computing Infrastructure. 1999

[3] K. Krauter, R. Buyya, and M. Maheswaran, A Taxonomy and Survey of Grid Resource Management Systems for Distributed Computing Softw. Pract, Exper. 2001;00:1-7

[4] I. Foster, C. Kesselman, and S. Tuecke The Anatomy of the Grid: Enabling Scalable Virtual Organizations, International Journal of High Performance Computing Applications, Vol.15, No.3, 2001, pp.200~222.

[5] I. Foster, C. Kesselman, G. Tsudik, and S.Tuecke, A Security Architecture for Computational Grids. In ACM Conference on Computers and Security, 1998, pp.83~91.

[6] I. Foster and C. Kesselman, Globus: A Metacomputing Infrastructure Toolkit, http://www.globus.org

[7] I. Foster and C. Kesselman, The Globus Project: A progress report, In Proceedings of the Heterogeneous Computing Workshop, 1998

[8] Status and Plans for Globus Toolkit™ 3.0(July 22, 2002), http://www.globus.org

[9] G. Laszewski, I. Foster, J. Gawor, CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids, Proceedings of the ACM java Grande Conference, 2000

[10] E. Gallopoulos, E. Houstis, and J. R. Rice. Problem-Solving Environments for Computational Science. IEEE Computational Science and Engineering, 1:11-23, 1994

[11] J. Novotny, The Grid Portal Development Kit, Concurrency: Pract. Exper. 2000; 00:1-7

[12] J. Boisseau, M. Dahan, S. Mock, K. Mueller, D. Sutton and M. Thomas, The Gridport Toolkit: A System for Building Grid Portals, Proc. of the 10th IEEE Intl. Symp. on High Perf. Dist. Comp 2001

[13] Wu Hong, Chi Xue-bin, Xu Feng, Creation of Web-based Interface for Supercomputing Environment, 5th Intl. conf. on Algorithms and Architectures for Parallel Processing

[14] Wu Hong, Chi Xue-bin, Hai-li Xiao, Sun-gen Deng, Creating of Computational Grid using Globus Toolkit and GPDK, 1st Workshop on Hardware/Software Support for Parallel and Distributed Scientific and Engineering Computing (SPDSEC-02), 2002

[15] I.Foster, N. Karonis, A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems, Proc. 1998 SC Conference, 1998

[16] GridFTP Update, January 2002, http://www-fp.globus.org/datagrid/gridftp.html

Collaborating components in mesh-based electronic packaging

P. Chow
Fujitsu Laboratories of Europe Limited
Hayes Park Central, Hayes End Road, Hayes,
Middlesex UB4 8FE, UK
E-mail: p.chow@fle.fujitsu.co.uk
And
C.-H. Lai
School of Computing and Mathematical Sciences
University of Greenwich
Old Royal Naval College, Park Row, Greenwich,
London SE10 9LS, UK
E-mail: C.H.Lai@gre.ac.uk

ABSTRACT

From the model geometry creation to the model analysis, the stages in between such as mesh generation are the most manpower intensive phase in a mesh-based computational mechanics simulation process. On the other hand the model analysis is the most computing intensive phase. Advanced computational hardware and software have significantly reduced the computing time – and more importantly the trend is downward. With the kind of models envisaged coming, which are larger, more complex in geometry and modelling, and multiphysics, there is no clear trend that the manpower intensive phase is to decrease significantly in time – in the present way of operation it is more likely to increase with model complexity. In this paper we address this dilemma in collaborating components for models in electronic packaging application.

1. INTRODUCTION

Computational aided engineering (CAE) is a key element in the design and analysis of electronic packaging products, from conceptual design to manufacturing operations and processes. The geometric model, commonly known as a computer aided design (CAD) model, facilitates all the down stream phases in the CAE process for a product. With the product design and analysis occur early in the sequence of the process chain, their success is vital to all subsequent activities to eventual product fabrication and time to market. For models needing computational mechanics analysis such as heat transfer of an integrated circuit chip, a model for the geometry of the solid object (and may be also the surrounding domain) is needed. This is frequently performed by the CAD system with built in solid modelling functions. Next the assigning of material types and boundary patches before finite element mesh generation of the model domain, then the analysis may begin. For a fully integrated CAE process the stages between the geometric modelling to the start of analysis is generally straightforward, but this is not always true. Some non-trivial examples are listed. (1) A particular kind of meshes is needed such as structured mesh solvers, which are not so common when the geometry modeller forms part of the analysis software package. (2) CAD model not directly suitable for mesh generation such as those assembled components that are not merged ideally leaving holes and gaps. (3) Mesh generation difficulties, including sheer size and complexity of the model, demanding on computing resources. Any problems encountered between the two stages require man-time for attention and adjustment, and these activities become frequent and are extremely time-consuming. Some of the difficulties can be solved by more advanced technology in algorithmic, software and hardware, but the kind of models envisaged which are much larger, more complex in geometry and modelling, and multiphysics, the man-time is unlikely to decrease significantly. It is possible to assume the total modelling time being consist of geometry creation, meshing and analysis. The analysis part may have greatly reduced its computational time, due to advances in computational hardware, software and numerical techniques, and more importantly the trend is downward. However the first two parts of the modelling time have not been reduced to the same level, because it is professional-manpower intensive, and with no clear downward trend. Therefore it is unlikely the total modelling time will decrease vastly in the present way of design operations.

The time to market in electronic packaging is decreasing rapidly, the product design and analysis cycle need to be shortened to meet the challenge, not just by reducing the number of design cycles but smart and more efficient ways of operations. In the sections below, the component meshing and gluing (CMG) approach is introduced for multi-chip module (MCM) problems in electronic packaging. This kind of problem is frequently makeup of basic shapes (blocks, cylinders, etc.) and when assembled together it becomes a complex geometry model to undertake; the relative scale between the components is a key factor. Figure 1 shows an example diagram of MCM geometry.

2. COMPONENT MESHING AND GLUING CONCEPT

The component meshing and gluing (CMG) approach takes a more natural approach, in the same light as an engineer assembling componentss in CAD to construct the solid model. Here, the object model is a collection of assembled meshed components. Like CAD systems' database that uses a parametric approach defining the geometry component relative to parameters such as length and thickness, for rapid model creation. The CMG follows the same concept with the component volume meshed, and is perhaps most suited to applications where models are constructed from a few basic shapes such as multi-chip module (MCM) models in electronic packaging.

The model of assembled components is then "glue" together by either merging components to form one mesh model or collaborating components using some iterative methods. The former methodology requires the use of polyhedral type elements to combine into one mesh model, and it is referred to as the CMG-Coupled strategy in this paper. One disadvantage

of the CMG-Coupled stratgey is that it does not apply to all solvers, for example, structured mesh solvers. Only solvers with polyhedral element capability can be considered. The latter methodology requires some iterative methods to collaborate the components through the exchange of boundary conditions between the components' interface, and it is referred to as the CMG-DDM in rest of paper. The solution of each component may be obtained by means of existing fast solvers. This is more universally applicable to all types of solvers, but one known disadvantage is that the computing time to solution is longer. Fast iterative methods in domain decomposition (thus we called this strategy CMG-DDM) can significantly shorten the time to solution but it is unlikely to match the single mesh case. For appropriate solvers, a combination of the two gluing strategies is possible.

The new approach virtually removed the difficulties in the previous operation and speedup the model creation and meshing processes. The tools for model creation and meshing in the process-chain are very much for constructing components for the meshed-component database. And any assembled models can be added to the database as meshed-components for use in other models. The volume-meshing element is not figured in the assembled model construction process, thus removing a potential manpower intensive element from the procedure. The only meshing related element that perhaps needing manpower input is the component's interface, this is not envisage – because it is a surface-meshing type problem and is one degree of dimension less than volume meshing thus full automation is expected. The downside in CMG-DDM is the expected longer analysis time, but if this is not too excessive the reduction time gained in model assembly could well offset the extra computing time and achieve a reduced overall modelling time. This is the ultimate goal, but until then when the analysis time is too excessive, we do have the parallel computing armaments to address the longer analysis time – remember the trend for the analysis part is downward.

3. NUMERICAL TECHNIQUES

Provided that the solvers can take polyhedral elements, the CMG-Coupled strategy does not require extra effort to put into the solvers. It is the mesh-model that needs to be connected at the finite-element mesh topology level, gluing the interfaces of the mesh components. This is essentially a finite-element mesh connectivity problem. In the CMG-DDM strategy, the domain decomposition method (DDM) [[1]] is ideally suited for the assembled-component model, with the non-overlapping class the most appropriate. A non-overlapping approach allows flexibility in the mesh processing, the methods of numerical solution, the handling of different physics, and the adoption of numerical solvers in each of the model components. This choice also makes the defect equation technique as developed in [[2]] an ideal method for CMG-DDM. The algorithmic methodology for the CMG-DDM by the defect equation techniques is as:

Let $Lu = f$ be defined in the domain Ω and $u = g$ on $\partial\Omega$, where L may be a nonlinear operator that depends on u, and g is a known function. The domain Ω is partitioned into M non-overlapped sub-domains such that $\bigcup_{i=1}^{M}\Omega_i = \Omega$ and $\Omega_i \bigcap \Omega_j = \phi$, for $i \neq j$. Each sub-domain is associated with a sub-model defined by $\mathbf{L}_i u_i = f_i$. The boundary of each sub-domain, $\partial\Omega_i$, subtracting the part of boundary which overlaps with the boundary of the entire problem is in essence a part of the interface. Therefore the interface, which attached to Ω_i, may be defined as $\gamma_i = \partial\Omega_i / \partial\Omega$. The boundary conditions defined on γ_i may be denoted by u_{γ_i} and it satisfies a defect equation, such as $D(u_\gamma) = 0$ [[2]]. Using superscripts to denote the number of gluing process, the CMG-DDM algorithm may be written as follows.

Initial values: $n = 0$; $u_i^{(0)} \quad i = 1,2,\cdots M$ are given.
Repeat { $n := n + 1$;
For $i = 1 \cdots M$ Do

$u_i^{(n)}$:= {Solve $L_i u_i^{(n)} = f_i$ in Ω_i subject to $u_i^{(n)} = g$ on $\partial\Omega \bigcap \partial\Omega_i$ and $u_i^{(n)}|_{\gamma_i} = u_{\gamma_i}$};

End-Do
Solve $D(u_\gamma) = 0$;}
Until Convergence achieved

When the model consists of a single domain (meshed component) then the For loop and the defect calculation, $D(u_\gamma) = 0$, are redundant, thereby reverting back to the conventional solution procedure. The CMG-Coupled cases are performed in this way. From the above algorithm the For loop may be run in parallel and on homogeneous computing systems the solution will be identical between parallel and scalar computations.

4. NUMERICAL EXPERIMENTS

The particular problem to be considered in this paper is governed by the 2-D energy equation, limited to conduction only, in temperature *u*. The variables in (2) are density (ρ), specific heat (c), thermal conductivity (k), time (t) and the source term (S).

$$\rho\, c\, \frac{\partial u}{\partial t} = \nabla \bullet (k\, \nabla u) + S(u) \qquad (2)$$

The nonlinearity is introduced in the form of a material phase-change in the source term. For solidification using the enthalpy source-based method this is given by

$$S(u) = L\,\rho\, \frac{\partial f(u)}{\partial t} \qquad (3)$$

where L is the latent heat and f is the liquid fraction. The algorithm for solving these kinds of problems may be found in papers by Chow & Cross [[3]] and Voller & Swaminathan [[4]] and is not discussed in this paper. Readers interested in obtaining more information are directed to these references.

A nonlinear problem with phase-change occurring inside the domain was conducted using the geometric model as depicted

in Figure 2. Figure 3 shows three different meshes used in the present tests. Figure 4 shows the cell invariant temperature distribution of the coupled computation and collaborating components. Due to the phase-change, latent heat energy is released and with both chip and board being cooler, heat transfer from connectors to both chip and board occurs. A contour plot of the temperature and liquid-fraction will show significant differences between coupled computation and collaborating components at the component interface – which is not right. This is because contour programs plot using vertex values. Since the cell-centred FVM has a value on the element centre interpolation is used to obtain value at the vertices. In the collaborating component the temperature is not correctly represented at the domain interfaces, whereas for the coupled the liquid-fraction is incorrectly represented. In this instant, the cell invariant result is the most appropriate to display.

The temperature plots are indistinguishable between the coupled computation and collaborating components. The liquid-fraction plots show a slight difference (within a cell width wide) which is more notable in the two coarser cases. Table 1 gives the total energy in the system with Mesh 1 being used as the reference guide towards accuracy and computing costs. The largest deviation of the solution from the referenced data is under 5% in the coarsest case and the times for coolaborating components are 4.7, 4.8 and 3.3 times more costly for the respective coupled mesh cases. Assuming 15% of the overall time is used for analysis, this implies the projected total modelling time of 11 for Mesh 1, which suggests there is some possibility for the collaborating component approach to be competitive.

5. SUMMARY

Numerical experiment relating to MCM models simulating solder solidification based on the enthalpy method [[3], [4]] shows the potential advantage of the method.

6. REFERENCES

[1] Domain Decomposition Methods in Sciences and Engineering, proceedings of international conference series on domain decomposition methods, published by DDM.org, URL: www.ddm.org

[2] C.-H. Lai, A.M. Cuffe and K.A. Pericleous, "A defect equation approach for the coupling of subdomains in domain decomposition methods", *Computers Math. Applic.*, **6** : 81 – 94, 1997

[3] P. Chow and M. Cross, "An enthalpy control volume-unstructured mesh (cv-um) algorithm for solidification by conduction only", *International Journal for Numerical Methods in Engineering,* **35** : 1849-1870, 1992

[4] V. Voller and C. Swaminathan, "General source-based methods for solidification phase change", *Numerical Heat Transfer*, **19** : 175-190, 1991

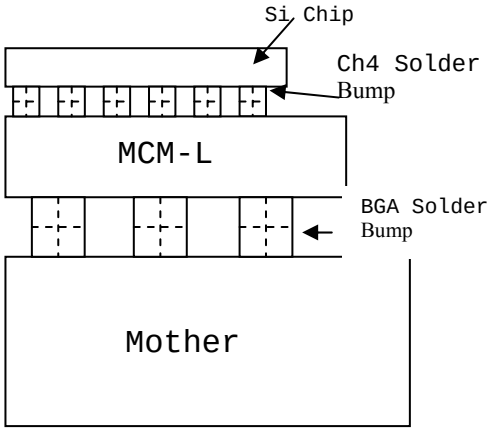


Figure 1 A multiple chip geometry

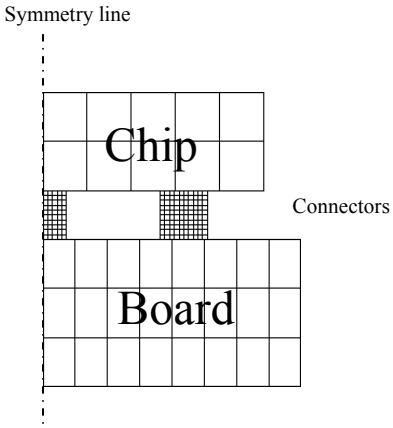


Figure 2 A chip board with connectors

Table 1 Total energy in the system

		Coupled computation	Collaborating components
Mesh 1	Total energy	4.5820×10^{9}	4.5820×10^{9}
	Relative error		1.925×10^{-7}
	CPU Time	1.65	7.77
Mesh 2	Total energy	4.6398×10^{9}	4.5853×10^{9}
	Relative error	1.261×10^{-2}	7.103×10^{-4}
	CPU Time	5.57	7.90
Mesh 3	Total energy	4.7892×10^{9}	4.7891×10^{9}
	Relative error	4.521×10^{-2}	4.716×10^{-2}
	CPU Time	4.19	5.39

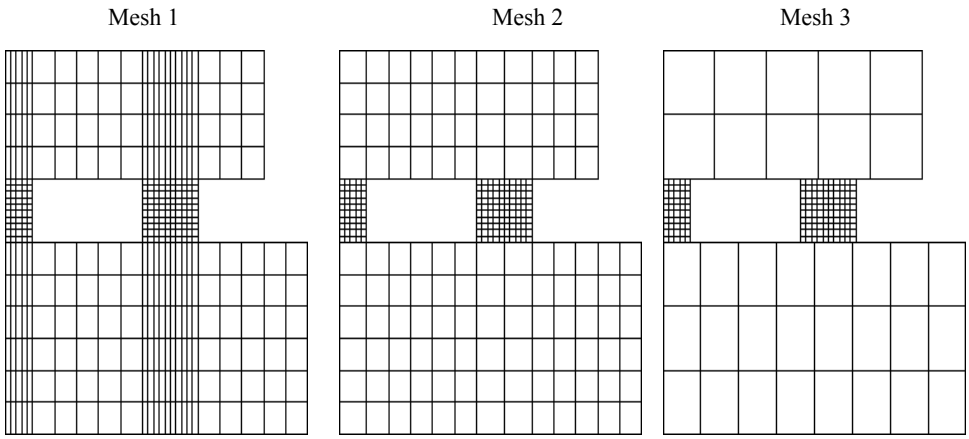


Figure 3 Three different meshes are used in the tests

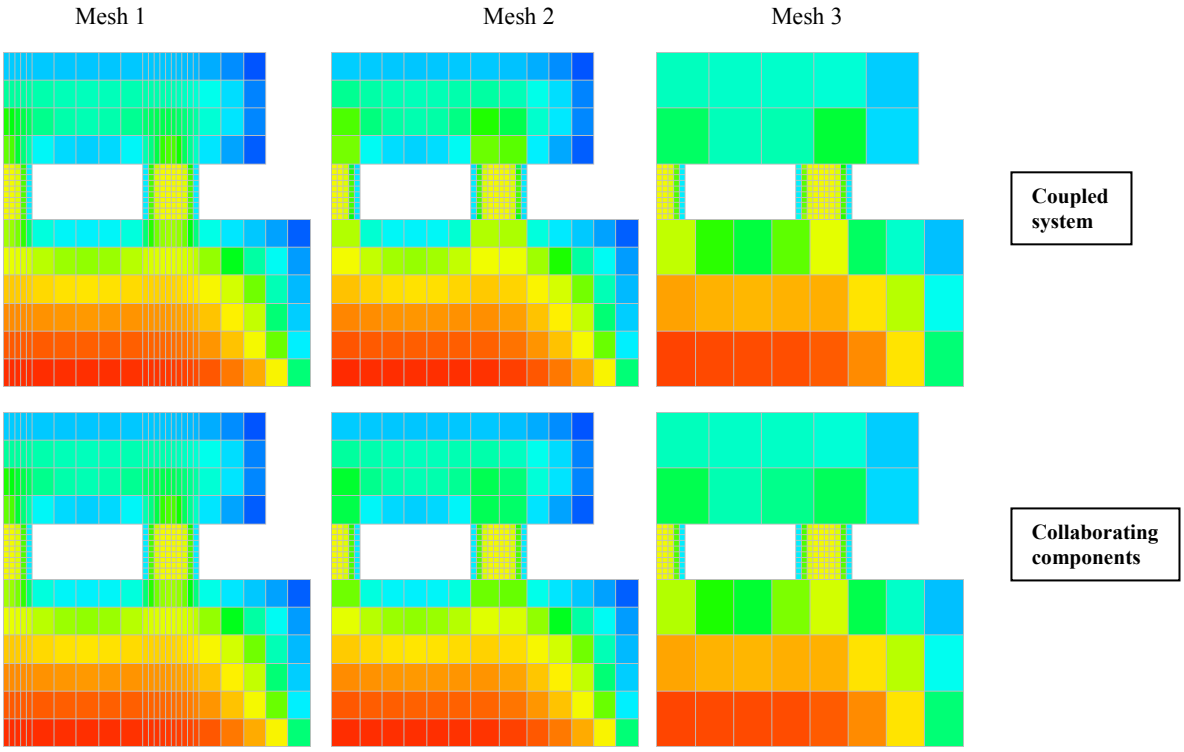


Figure 4 Comparison of the cell invariant temperature distribution

Designing EDIFACT Message Structures with XML Schema

Qianxing Xiong
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail:qxxi@ public.wh.hb.cn
Bo Meng
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail:mengbo@263.net.cn
Xinming Tan
College of Computer Science and Technology
Wuhan University of Technology
Wuhan 430063 P. R. China
E-mail:tanxming@public.wh.hb.cn

ABSTRACT

With the development of applications of XML and the publication of XML schema recommendation by W3C, XML+XMLschema-based EDIFACT is becoming a good solution to the electronic commerce applications. Now a lot of large-scale enterprises want to make the traditional electronic commerce systems migrated to XML schema-based EDIFACT. At the same time many small and medium-scale enterprises want to implement electronic commerce systems with XML schema-based EDIFACT. The design and implementation of EDIFACT message structures based on XML schema is the most important work and the base of applying XML +XML schema-based EDIFACT to electronic commerce systems and implementations of new electronic commerce systems. In this paper we present a method of design and implementation of EDIFACT message structures based on XML schema and illustrate it with examples.

Keywords: electronic commerce, XML, EDIFACT, XML schema, message structure, datatype.

1. INTRODUCTION

In the past years a lot of electronic commerce systems were based on EDIFACT [1] (Electronic Data Interchange For Administration, Commerce and Transport), which is an international EDI (Electronic Data Interchange) standard developed by the United Nations. This international standard includes the rules on the application level for the structuring of user data and of associated service data in the interchange of messages in an open environment. Beside the syntax the EDIFACT standard [2] covers also the definition of data elements (the data information as basic component for message types), segments (functionally related sets of data elements) and message types (structured representation of the full information on an electronic commerce transaction).
Unfortunately, only those large-sized enterprises can afford the expensive electronic commerce systems based on EDIFACT. The small and medium scale enterprises cannot bear the cost of implementation and the difficulty of development. This cost includes the substantial investment in legacy information system, managements, maintenance and software etc [3]. At the same time EDIFACT has been criticized for poor design, confusing or absent semantics [4]. Those difficulties block the implementation and generalization of electronic commerce system based on EDIFACT.
The emergence of XML [5] (eXtensible Markup Language) resolves those problems. In 1996 W3C (World Wide Web Consortium) joined with SGML (Standard Generalized Markup Language) experts to form an SGML Working Group, which strategically pruned SGML into a refined subset now known as XML, which, published in 1997, is a metalanguage and can be as the standard for self-describing data exchange in Internet applications. XML makes electronic commerce system developed rapidly and maintained easily. Now XML has become the first choice in the field of defining data interchange formats in electronic commerce. In May 2001 W3C published a recommendation of XML Schema [6,7,8.] XML schema can be used to describe the structure of XML document and define the semantics of element.
Thus we can use XML+XML schema-based EDIFACT as the solution to electronic commerce applications. In order to develop electronic commerce system based on XML schema-based EDIFACT, the first work is to use the XML schema to describe the EDIFACT message structure, which is a very important work and the base of applying XML to electronic commerce system based on EDIFACT and of implementing new electronic commerce systems.
This paper is organized as follows. In section 2, we discuss the related work. In section 3, we describe the message structure of EDIFACT. In section 4, we present how to specify EDIFACT message structure with XML schema. The section 5 concludes our work.

2. RELATED WORK

There have been a lot of papers and projects discussing the XML/EDI owing to the birth of XML and XML schema. But until now there is no discussion on using XML schema to define EDIFACT.
The CEN/ISSS Electronic Commerce Workshop published Ref No CWA 14162:Datatyping for Electronic Data Interchange in March 2000 [9]. This document concentrates on techniques for defining and constraining data or code set values used within B2B (business-to-business) electronic data interchange messages. The document only discusses the several datatyping for Electronic Data Interchange and don't give the EDIFACT message structure defined by XML schema. Multek Sweden AB developed IGML [10] in February 2000. It uses DTD to describe the EDIFACT

message structure.
Michael Koehne also used the DTD to describe the EDIFACT in 2000[11].
Due to the shortage of DTD, the DTD-specified EDIFACT is not completely consistent to EDIFACT. For example, we cannot use DTD to define datatype we need and constrain the element content according to the special application

3. EDIFACT

Since 1988 the United Nations has been developing the EDIFACT to meet the requirements of an internationally valid general business standard. This international standard includes the rules on the application level for the structuring of user data and of associated service data in the interchange of messages in an open environment. Beside the syntax, the EDIFACT standard covers also the definition of data elements (the data information as basic component for message types), segments (functionally related sets of data elements), and message types (structured representation of the full information on an electronic business transaction).
EDIFACT can be considered as a dynamic standard since new message types are developed and definitions of existing message types are be changing in the course of time. The complete documentation of the EDIFACT guidelines is included in an UN/EDIFACT directory (Fig.1 UN/EDIFACT directory), which comprises the message type directory, the segment type directory, the composite data element type directory, the simple data element type directory, and the code list directory.

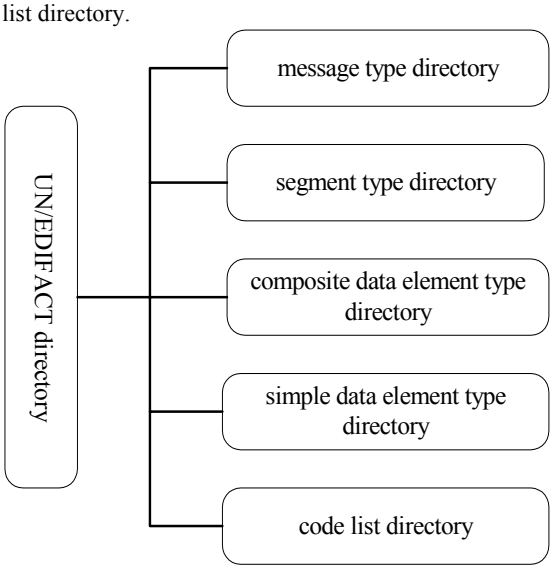


Fig.1 UN/EDIFACT directory

EDIFACT message comprises an ordered set of segments. Segments may be grouped. A segment group comprises an ordered set of segments: a trigger segment and at least one more segment or segment group. The trigger segment shall be the first segment in the segment group, shall have a status of mandatory and a maximum number of occurrences of one. A segment comprises an ordered set of stand-alone data elements and/or composite data elements, each of which are permitted to repeat, if so stated in the segment specification. A composite data element comprises an ordered set of two or more component data elements. A simple data element contains a single data element value. A simple data element is used either as a stand-alone data element or as a component data element. A stand-alone data element occurs in a segment outside a composite data element. A component data element occurs within a composite data element. The message structure of CALINF is as follows:

UNH Message header	M	1
BGM Beginning of message	M	1
DTM Date/time/period	C	9
Segment group 1	C	99
FTX Free text	M	1
MEA Measurements	C	9
EQN Number of units	C	1
Segment group 2	C	9
RFF Reference	M	1
DTM Date/time/period	C	9
Segment group 3	M	9
NAD Name and address	M	1
Segment group 4	C	9
CTA Contact information	M	1
COM Communication contact	C	9
Segment group 5	M	1
TDT Details of transport	M	1
DTM Date/time/period	C	9
RFF Reference	C	9
Segment group 6	M	9
LOC Place/location identification	M	1
DTM Date/time/period	C	9
DIM Dimensions	C	1
FTX Free text	C	9
Segment group 7	C	9
QTY Quantity	M	1
FTX Free text	C	1
UNT Message trailer	M	1

Fig.2 the message structure of CALINF

4. DESIGNING EDIFACT MESSAGE STRUCTURES WITH XML SCHEMA

The EDIFACT message structures documentation (Fig.3 the documentation structure of EDIFACT message structures based on XML schema) we developed comprises simpleEDType.xsd, simpleEDeclaANDcompositeEDType.xsd, compositeEDeclaANDsegmnetEDtype.xsd, segmentEDecla.xsd, and the message documents that include the declarations of segment group element and message and the definitions of segment group element datatype and message datatype. simpleEDType.xsd includes the definitions of simple element datatype of EDIFACT. simpleEDeclaANDcompositeEDType.xsd includes the definitions of composite element datatype of EDIFACT and declarations of simple element of EDIFACT. compositeEDeclaANDsegmnetEDtype.xsd includes the declarations of composite element datatype of EDIFACT and the definitions of segment element datatype of EDIFACT. segmentEDecla.xsd includes the declarations of the segment

element of EDIFACT. Each message structure is a document that includes the declarations of segment group element and message and the definitions of segment group element datatype and message datatype.

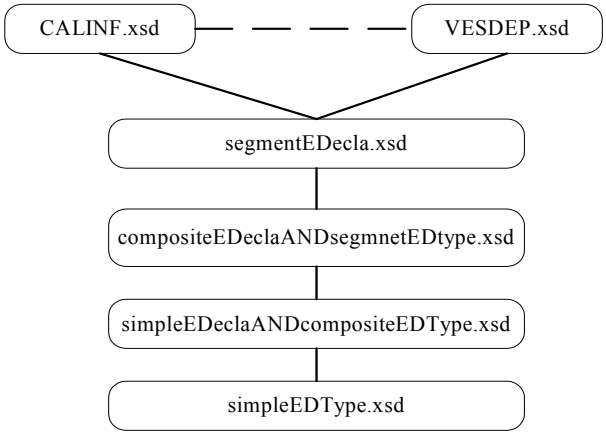


Fig.3 the documentation structure of EDIFACT message structures based on XML schema

When we describe the EDIFACT message structure with XML schema, the most important issue is defining the element datatypes based on EDIFACT.

The XML schema specification defines the following three kinds of datatypes: Primitive datatypes (string, binary, etc.), Derived datatypes (CDATA, token, etc), User-derived datatypes. The User-derived datatypes allow users to create complex datatypes that are composed of sets of primitive datatypes. User-derived datatypes can include enumerated lists of values, which can include values of different datatypes. For the string data type, users can define patterns that the string must conform to. For numeric values, maximum and minimum values can be specified (inclusively or exclusively), as scale and precision. Booleans can be represented as true or 0 and false or 1. Dates and time can be expressed using various ISO 8601-based formats. Datatypes can also be derived as the union of two other datatypes, as lists of values conforming to another datatype, or as restrictions on an existing datatype.

We take the simple element 5243 as an example to show the definition of element datatypes. We know that the value of element 5243 comprises A, B, C, D, E, F, K, M, N, Q, R and S. So we make the base datatype of the datatype of 5243 enumeration datatype. The dataType of 5243 is s5243simpleElementDataType, which is a complexType. At the same time we define the attributes of anno, flag, code, desc, repr and posi for every simple element in order to be understood easily by the developer. The element 5243 is defined with XML schema as follows:

```
- <xs:complexType
   name="s5243simpleElementDataType
   ">
    - <xs:simpleContent>
       -              <xs:extension
          base="s5243enumeration
          Type">
              <xs:attributeGroup
             ref="SimpleEleme
             ntAttributeGr" />
       </xs:extension>
  </xs:simpleContent>
</xs:complexType>
 - <xs:simpleType
    name="s5243enumerationType">
     - <xs:restriction base="xs:string">
                  <xs:enumeration
          value="A" />
                  <xs:enumeration
          value="B" />
                  <xs:enumeration
          value="C" />
                  <xs:enumeration
          value="D" />
                  <xs:enumeration
          value="E" />
                  <xs:enumeration
          value="F" />
                  <xs:enumeration
          value="K" />
                  <xs:enumeration
          value="M" />
                  <xs:enumeration
          value="N" />
                  <xs:enumeration
          value="Q" />
                  <xs:enumeration
          value="R" />
                  <xs:enumeration
          value="S" />
   </xs:restriction>
</xs:simpleType>
 - <xs:attributeGroup
    name="SimpleElementAttributeGr">
        <xs:attribute     name="anno"
      type="xs:string"  use="required"
      />
        <xs:attribute     name="flag"
      type="xs:string"  use="optional"
      />
        <xs:attribute     name="code"
      type="xs:integer"
      use="required" />
        <xs:attribute     name="desc"
      type="xs:string"  use="optional"
      />
        <xs:attribute     name="repr"
      type="xs:string"  use="required"
      />
        <xs:attribute     name="posi"
      type="xs:integer"
      use="optional" />
</xs:attributeGroup>
```

Now we can declare 5243 as follows:

```
<xs:element              name="s5243"
      type="s5243simpleElementData
      Type" />
```

In succession we show how to define the datatypes of composite elements and declare composite elements. We take an example for the composite element c082 to show the definition of element datatypes. C082 is described according to the definition of EDIFACT as follows:

C082 PARTY IDENTIFICATION DETAILS
Desc: Identification of a transaction party by code.
010 3039 Party identifier M an..35
020 1131 Code list identification code C an..17
030 3055 Code list responsible agency code C an..3

First we define its datatype:

```
- <xs:complexType
      name="c082compositeElement
      DataType">
   - <xs:sequence>
      <xs:element ref="s3039" />
          <xs:element   ref="s1131"
      minOccurs="0" />
          <xs:element   ref="s3055"
      minOccurs="0" />
      </xs:sequence>
                <xs:attributeGroup
      ref="CompositeElementAttribut
      eGr" />
   </xs:complexType>
    <xs:attributeGroup
      name="CompositeElementAttri
      buteGr">
          <xs:attribute
      name="anno"   type="xs:string"
      use="required"/>
          <xs:attribute
      name="code"  type="xs:integer"
      use="required"/>
          <xs:attribute
      name="desc"   type="xs:string"
      use="optional"/>
          <xs:attribute
      name="posi"   type="xs:integer"
      use="optional"/>
    </xs:attributeGroup>
```

Then we can declare c082 as follows:

```
<xs:element                          name="c082"
type="c082compositeElementDataType" />
```

We take an example for segment element RFF to show the definition of segment element datatypes. RFF is described according to the definition of EDIFACT as follows:

```
RFF   REFERENCE
      Function: To specify a reference.
010                  C506   REFERENCE
M   1
```

First we define its datatype:

```
<xs:complexType name="RFFsegmentDataType">
      <xs:sequence>
          <xs:element ref="c506"/>
      </xs:sequence>
      <xs:attributeGroup
ref="SegmentAttributeGr"/>
</xs:complexType
```

We define attributes to describe segment element:

```
-          <xs:attributeGroup
      name="SegmentAttributeGr">
          <xs:attribute  name="abbr"
      type="xs:string"  use="optional"
      />
          <xs:attribute  name="anno"
      type="xs:string"  use="required"
      />
          <xs:attribute  name="desc"
      type="xs:string"  use="optional"
      />
</xs:attributeGroup>
```

Then we can declare RFF as follows:

```
<xs:element          name="RFF"
      type="RFFsegmentDataType" />
```

We can define other simple element, composite element, segment element and segment group element of EDIFACT with the method we introduced before.

In the last we can get the EDIFACT message structure based on XML schema. For example, the CALINF message structure is in CALINF.xsd. It's content is partly shown as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.4
(http://www.xmlspy.com) by myh (company) -->
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<xs:include
schemaLocation="http://xml.whut.edu.cn/xmledi/segmentDecl
aration.xsd"/>
<xs:element name="CALINF"
type="MessageCalinfDataType"/>
<xs:element name="SegmentGroup1"
type="CalinfSegmentGr1DataType"/>
<xs:element name="SegmentGroup2"
type="CalinfSegmentGr2DataType"/>
<xs:element name="SegmentGroup3"
type="CalinfSegmentGr3DataType"/>
<xs:element name="SegmentGroup4"
type="CalinfSegmentGr4DataType"/>
<xs:element name="SegmentGroup5"
type="CalinfSegmentGr5DataType"/>
<xs:element name="SegmentGroup6"
type="CalinfSegmentGr6DataType"/>
<xs:element name="SegmentGroup7"
type="CalinfSegmentGr7DataType"/>
<xs:complexType
name="MessageCalinfDataType">
      <xs:sequence>
          <xs:element ref="UNH"/>
          <xs:element ref="BGM"/>
          <xs:element ref="DTM"
minOccurs="0" maxOccurs="9"/>
          <xs:element ref="SegmentGroup1"
minOccurs="0" maxOccurs="99"/>
          <xs:element ref="SegmentGroup2"
minOccurs="0" maxOccurs="9"/>
          <xs:element ref="SegmentGroup3"
maxOccurs="9"/>
          <xs:element ref="SegmentGroup5"/>
          <xs:element ref="SegmentGroup7"
minOccurs="0" maxOccurs="9"/>
          <xs:element ref="UNT"/>
      </xs:sequence>
</xs:complexType>
</xs:schema>
```

5. CONCLUSION

With the development of electronic commerce and application of XML and the publication of XML schema recommendation, a lot of large scale enterprises want to make electronic commerce systems migrate from traditional EDIFACT to XML schema-based EDIFACT. At the same time many SMEs want to implement electronic commerce system with XML schema-based EDIFACT. XML+XML schema-based EDIFACT is a good solution to the electronic commerce applications. The XML document includes the message information while the XML schema can be used to define the

message structure and datatypes of elements according to EDIFACT, so we can use the XML schema to validate the XML documents and make them conform to the specification of the XML schema –based EDIFACT. In this paper we use XML schema to define EDIFACT, which is the base of generalization of electronic commerce systems based on XML schema-based EDIFACT.

The documents we developed comprise simpleEDType.xsd, simpleEDeclaANDcompositeEDType.xsd, compositeEDeclaANDsegmnetEDtype.xsd, segmentEDecla.xsd, and the documents that include the declarations of segment group elements and message and the definitions of segment group element datatypes and message datatype. The documentation structure is consistent to the EDIFACT directory and is easy to be understood. Dividing definitions of simple elements, composite elements, segments, segment group into different documents makes them reusable and easy to apply.

Till now we have defined about twenty message structures with the method presented in our paper. We will go on this work until all EDIFACT message structures be specified with XML schema.

6. REFERENCES

[1] EDIFACT, http://www.unece.org/trade/untdid.

[2] ISO 9735-1: Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4) —Part 1:Syntax rules common to all parts, together with syntax service directories for each of the parts.

[3] Steven O. Kimbrough, "EDI, XML, and Seeing through Transparency", AspenWorld 2000, Orlando, FL, February 8, 2000.

[4] Nabil R.Adam and Yesha, editors, Electronic Commerce: Current Research Issues and Applications, Volume 1028 of Lecture Notes in computer Science. Springer, Berlin, Germany, 1996.

[5] Extensible Markup Language (XML) 1.0 (Second Edition), http://www.w3.org/TR/2000/REC-xml-20001006, W3C Recommendation 6 October 2000.

[6] XML Schema Part 0:Primer, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/, W3C Recommendation, 2 May 2001.

[7] XML Schema Part 1:Structures, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/, W3C Recommendation, 2 May 2001.

[8] XML Schema Part 2:Datatypes, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/, W3C Recommendation, 2 May 2001

[9] CEN/ISSS Workshop Agreement: CWA 14162 Datatyping for Electronic Data Interchange, March 2001

[10] http://www.multek.com

[11] http://www.xml-edifact.org

[12] Steven O. Kimbrough, "EDI, XML, and the Myth of Semantic Transparency," Bloomington, IN, October 23, 1999

OpenMP Parallel Implementation of Finite Difference Time Domain Electromagnetism Application

Yi Pan and Ying Luo
Department of Computer Science
Georgia State University
University Plaza
Atlanta, GA30303
E-mail: pan@cs.gsu.edu
Minyi Guo*
Department of Computer Software
The University of Aizu
Aizu-Wakamatsu City, Fukushima 965-0001, Japan
E-mail: minyi@u-aizu.ac.jp
Joseph W. Haus
Electro-Optics Program
The University of Dayton
Dayton, OH45469-0245
E-mail: Joseph.Haus@notes.udayton.edu
Majeed M. Hayat
Department of Electrical & Computer Engineering
The University of New Mexico
Albuquerque, NM87131-1356
E-mail: hayat@eece.unm.edu

ABSTRACT

We report the OpenMP parallel implementation of a finite difference time domain (FDTD) method for computational electrodynamics. We have identified several time-intensive procedures in the program and parallelized the major loops within them after careful examination. Different loop scheduling schemes have been used and tested in order to reduce computation times. The final parallelized version speeds up the computation by nearly a factor of four between the single processor and eight processor test runs on an SGI Origin 2000 parallel system. The speedup plateaus after eight CPUs, but we expect better scalability will be achieved if larger problem sizes are used. Besides the advantage of reduced execution times, our parallel program can also solve FDTD problems of much larger sizes than the sequential code due to much larger memory space available to us on parallel systems.

1. INTRODUCTION

In order to keep up with the demand for higher bandwidth imposed by the growth of the Internet, rapid development of new fiber-optic systems is needed. The heart of these systems is photonic devices (lasers, detectors, modulators, switches, etc.) that can relieve the bottlenecks imposed by their predecessors [Grote and Venghaus, 2001]. Scientists and engineers are engaged in a race to continually make device improvements that will meet the future demands of the communications industry. Furthermore, the complexity of the new devices requires the development of reliable modeling techniques to optimize the device performance and shorten the design cycle.

One of the most promising of the novel device concepts is the introduction of periodic dielectric materials called photonic band gap (PBG) structures or photonic crystals into the design [Weisbuch 2000]. PBG structures have boundaries spaced less than a wavelength apart; this places stringent demands on the memory storage and the computation time [Haus 1994]. Most commercial computer aided design software cannot be reliably employed to simulate realistic device designs.

The Finite Difference Time Domain (FDTD) method is an order N method to solve Maxwell's equations. It is well suited to applying paralellization techniques to significantly reduce the computation time and sharing the memory storage. One version of the FDTD algorithm written in Fortran 90 was developed by Ward and coworders [Ward 1998]. Their code performs two basic types of calculations. The first is a band structure calculation, which starts off with an arbitrary set of initial fields corresponding to some wavevector K, integrates in time storing the fields at some random sampling points and then Fourier transforms into the frequency domain. The peaks in the frequency spectrum correspond to the frequencies the the allowed modes. The second type of calculation finds the time dependent Green's function by setting the initial fields to be a delta function in space for one of the field components, zero for all the others. The fields are integrated in time and stored at each time step to give $g(t, r, r')$ Fourier transforming gives $G(\omega, r, r')$.

2. BACKGROUND THEORY

Discrete versions of Maxwell's equations used as the basis for a finite difference time domain calculation of a form similar to that of Yee [Taflove 1995]. We begin with Maxwell's equations in generalized coordinate form

$$\nabla_q \times \hat{H} = \varepsilon_0 \hat{\varepsilon}(r) \frac{\partial \hat{E}}{\partial t}$$

$$\nabla_q \times \hat{E} = -\mu_0 \hat{\mu}(r) \frac{\partial \hat{H}}{\partial t}$$

The renormalised fields are,

$$\hat{E}_i = Q_i E_i \qquad \hat{H}_i = Q_i H_i$$

with,

$$Q_i = \sqrt{\left(\frac{\partial x}{\partial q_i}\right)^2 + \left(\frac{\partial y}{\partial q_i}\right)^2 + \left(\frac{\partial z}{\partial q_i}\right)^2}$$

The effective permittivity $\hat{\varepsilon}$ and permeability $\hat{\mu}$ can be shown to be tensors of the following form,

$$\hat{\varepsilon}^{ij}(r) = \varepsilon(r) g^{ij} \left| u_1 \cdot u_2 \times u_3 \right| \frac{Q_1 Q_2 Q_3}{Q_i Q_j}$$

$$\hat{\mu}^{ij}(r) = \mu(r) g^{ij} \left| u_1 \cdot u_2 \times u_3 \right| \frac{Q_1 Q_2 Q_3}{Q_i Q_j}$$

Where the vectors μ_i are unit vectors pointing along the axes of the generalised co-ordinate system. The point of this result is that all the details of the co-ordinate system have been hidden inside the effective ε and μ. We use $\hat{\varepsilon}^{ij}$ and $\hat{\mu}^{ij}$ at each point.

The discrete versions of these time dependent equations are

$$\frac{\partial \hat{E}}{\partial q_i} \mapsto \Delta_i^+ \hat{E} = \hat{E}(r + Q_i, t) - \hat{E}(r, t)$$
$$\frac{\partial \hat{H}}{\partial q_i} \mapsto \Delta_i^- \hat{H} = \hat{H}(r, t) - \hat{H}(r - Q_i, t)$$

and

$$\frac{\partial \hat{E}}{\partial t} \mapsto \frac{\Delta_r^+ \hat{E}}{\delta t} = \frac{\hat{E}(r, t + \delta t) - \hat{E}(r, t)}{\delta t}$$
$$\frac{\partial \hat{H}}{\partial q_i} \mapsto \frac{\Delta_r^- \hat{H}}{\delta t} = \frac{\hat{H}(r, t) - \hat{H}(r, t - \delta t)}{\delta t}$$

Where $Q_i = Q_i u_i$. We can use the discrete spacial derivatives to define a discrete version of the curl,

$$\left[\nabla_q^+ \times F\right]_3 = \Delta_1^+ F_2 - \Delta_2^+ F_1$$

etc. Substituting into Maxwell's equations gives,

$$\nabla_q^+ \times \hat{E} = -\frac{\mu_0}{\partial t} \hat{\mu}(t) \Delta_r^+ \hat{H}$$
$$\nabla_q^- \times \hat{H} = +\frac{\varepsilon_0}{\partial t} \hat{\varepsilon}(r) \Delta_r^- \hat{E}$$

Here we should note that on the discrete mesh the Q_i factors simply correspond to the spacings between the mesh points in each direction.

This choice of finite differences is equivalent to making the following approximations to k and ω. For the electric field terms,

$$k_j \mapsto k_j^{\mathrm{II}} = \frac{\exp\left[ik_j Q_j\right] - 1}{iQ_j}$$
$$\omega \mapsto \omega^{\mathrm{II}} = \frac{\exp\left[-i\omega\delta t\right] - 1}{-i\delta t}$$

And for the magnetic field terms,

$$k_j \mapsto k_j^{\mathrm{II}} = \frac{1 - \exp\left[-ik_j Q_j\right]}{iQ_j}$$
$$\omega \mapsto \omega^{\mathrm{II}} = \frac{1 - \exp\left[i\omega\delta t\right]}{-i\delta t}$$

For the free-space dispersion relation, $\omega / k = c_0$ these approximations give,

$$\frac{4}{\delta t^2} \sin^2\left(\frac{\omega \delta t}{2}\right) = 4c_0^2$$
$$\left\{\frac{1}{Q_1^2} \sin^2\left(\frac{Q_1 k_x}{2}\right) + \frac{1}{Q_2^2} \sin^2\left(\frac{Q_2 k_y}{2}\right) + \frac{1}{Q_3^2} \sin^2\left(\frac{Q_3 k_z}{2}\right)\right\}$$

For reasons of numerical stability it is desirable to make the electric and magnetic fields of roughly equal magnitude. This can be done by introducing a rescaled magnetic field.

$$\hat{H}' = \frac{\delta t}{\varepsilon_0 Q_0} \hat{H}$$

Where we introduce the constant Q_0 to have the dimensions of length and be roughly equal in length to the Q_i' s. If we also introduce a rescaling to $\hat{\varepsilon}$ and $\hat{\mu}$ then Maxwell's equations become,

$$\nabla_r^+ \times \hat{E}(r, t) = \left[\hat{\varepsilon}(r)\right]^{-1} \nabla_q^- \times \hat{H}'(r, t)$$
$$\nabla_r^- \hat{H}'(r, t) = -\left\{\frac{\delta t c_0}{Q_0}\right\}^2 \left[\hat{u}(r)\right]^{-1} \nabla_q^+ \times \hat{E}(r, t)$$

where $\hat{\varepsilon}$ and $\hat{\mu}$ are now,

$$\hat{\varepsilon}^{ij}(r) = \varepsilon(r) g^{ij} \left| u_1 \cdot u_2 \times u_3 \right| \frac{Q_1 Q_2 Q_3}{Q_i Q_j Q_0}$$
$$\hat{\mu}^{ij}(r) = \mu(r) g^{ij} \left| u_1 \cdot u_2 \times u_3 \right| \frac{Q_1 Q_2 Q_3}{Q_i Q_j Q_0}$$

So if we know the electric and magnetic fields at time *t* we can calculate the fields at $t + \delta t$

$$\hat{E}_1(r, t + \delta t) = \hat{E}_1(r, t)$$
$$+ \left[\hat{\varepsilon}^{-1}(r)\right]^{11} \left\{\hat{H}_3'(r, t) - \hat{H}_3'(r - b, t) - \hat{H}_2'(r, t) + \hat{H}_2'(r - c, t)\right\}$$
$$+ \left[\hat{\varepsilon}^{-1}(r)\right]^{12} \left\{\hat{H}_1'(r, t) - \hat{H}_1'(r - c, t) - \hat{H}_3'(r, t) + \hat{H}_3'(r - a, t)\right\}$$
$$+ \left[\hat{\varepsilon}^{-1}(r)\right]^{13} \left\{\hat{H}_2'(r, t) - \hat{H}_2'(r - a, t) - \hat{H}_1'(r, t) + \hat{H}_1'(r - b, t)\right\}$$

$$\hat{E}_2(r, t+\delta t) = \hat{E}_2(r, t)$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{21}\left\{\hat{H}'_3(r, t) - \hat{H}'_3(r-b, t) - \hat{H}'_2(r, t) + \hat{H}'_2(r-c, t)\right\}$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{22}\left\{\hat{H}'_1(r,t) - \hat{H}'_1(r-c,t) - \hat{H}'_3(r,t) + \hat{H}'_3(r-a,t)\right\}$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{23}\left\{\hat{H}'_2(r,t) - \hat{H}'_2(r-a,t) - \hat{H}'_1(r,t) + \hat{H}'_1(r-b,t)\right\}$$

$$\hat{E}_3(r,t+\delta t) = \hat{E}_3(r,t)$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{31}\left\{\hat{H}'_3(r,t) - \hat{H}'_3(r-b,t) - \hat{H}'_2(r,t) + \hat{H}'_2(r-c,t)\right\}$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{32}\left\{\hat{H}'_1(r,t) - \hat{H}'_1(r-c,t) - \hat{H}'_3(r,t) + \hat{H}'_3(r-a,t)\right\}$$
$$+\left[\hat{\varepsilon}^{-1}(r)\right]^{33}\left\{\hat{H}'_2(r,t) - \hat{H}'_2(r-a,t) - \hat{H}'_1(r,t) + \hat{H}'_1(r-b,t)\right\}$$

$$\hat{H}'_1(r,t+\delta t) = \hat{H}'_1(r,t) - \left(\frac{\delta t c_0}{Q_0}\right)^2 \times$$
$$[\ +\left[\hat{\mu}^{-1}(r)\right]^{11}\left\{\hat{E}_3(r+b,t) - \hat{E}_3(r,t) - \hat{E}_2(r+c,t) + \hat{E}_2(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{12}\left\{\hat{E}_1(r+c,t) - \hat{E}_1(r,t) - \hat{E}_3(r+a,t) + \hat{E}_3(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{13}\left\{\hat{E}_2(r+a,t) - \hat{E}_2(r,t) - \hat{E}_1(r+b,t) + \hat{E}_1(r,t)\right\}\ ]$$

$$\hat{H}'_2(r,t+\delta t) = \hat{H}'_2(r,t) - \left(\frac{\delta t c_0}{Q_0}\right)^2 \times$$
$$[\ +\left[\hat{\mu}^{-1}(r)\right]^{21}\left\{\hat{E}_3(r+b,t) - \hat{E}_3(r,t) - \hat{E}_2(r+c,t) + \hat{E}_2(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{22}\left\{\hat{E}_1(r+c,t) - \hat{E}_1(r,t) - \hat{E}_3(r+a,t) + \hat{E}_3(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{23}\left\{\hat{E}_2(r+a,t) - \hat{E}_2(r,t) - \hat{E}_1(r+b,t) + \hat{E}_1(r,t)\right\}\ ]$$

$$\hat{H}'_3(r,t+\delta t) = \hat{H}'_3(r,t) - \left(\frac{\delta t c_0}{Q_0}\right)^2 \times$$
$$[\ +\left[\hat{\mu}^{-1}(r)\right]^{31}\left\{\hat{E}_3(r+b,t) - \hat{E}_3(r,t) - \hat{E}_2(r+c,t) + \hat{E}_2(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{32}\left\{\hat{E}_1(r+c,t) - \hat{E}_1(r,t) - \hat{E}_3(r+a,t) + \hat{E}_3(r,t)\right\}$$
$$+\left[\hat{\mu}^{-1}(r)\right]^{33}\left\{\hat{E}_2(r+a,t) - \hat{E}_2(r,t) - \hat{E}_1(r+b,t) + \hat{E}_1(r,t)\right\}\ ]$$

where $a \equiv Q_1$ $b \equiv Q_2$ and $c \equiv Q_3$.

Stability criterion can be found as follows. The discretized equations give a stable updating procedure for the fields if the time step is kept sufficiently small. The criterion is then easy to find. Starting from the dispersion on the discrete mesh,

$$\frac{4}{\delta t^2}\sin^2\left(\frac{\omega \delta t}{2}\right) = 4c_0^2$$
$$\left\{\frac{1}{Q_1^2}\sin^2\left(\frac{Q_1 k_x}{2}\right) + \frac{1}{Q_2^2}\sin^2\left(\frac{Q_2 k_y}{2}\right) + \frac{1}{Q_3^2}\sin^2\left(\frac{Q_3 k_z}{2}\right)\right\}$$

The condition that the maximum value of the right hand side must correspond to a real frequency gives,

$$(\delta t)^2 < \left(\frac{c_0^2}{Q_1^2} + \frac{c_0^2}{Q_2^2} + \frac{c_0^2}{Q_3^2}\right)^{-1}$$

3. OPENMP PARALLEL PROGRAMMING

The rapid and widespread acceptance of shared-memory multiprocessor architectures has created a pressing demand for an efficient way to program these systems. At the same time, developers of technical and scientific applications in industry and in government laboratories find they need to parallelize huge volumes of code in a portable fashion. The OpenMP Application Program Interface (API) [Chandra 2000] supports multi-platform shared-memory parallel programming in C/C++ and Fortran on all architectures, including Unix platforms and Windows NT platforms. Jointly defined by a group of major computer hardware and software vendors, OpenMP is a portable, scalable model that gives shared-memory parallel programmers a simple and flexible interface for developing parallel applications for platforms ranging from the desktop to the supercomputer. It consists of a set of compiler directives and library routines that extend FORTRAN, C, and C++ codes to express shared-memory parallelism.

OpenMP's programming model uses fork-join parallelism: master thread spawns a team of threads as needed. Parallelism is added incrementally: i.e. the sequential program evolves into a parallel program. Hence, we do not have to parallelize the whole program at once. OpenMP is usually used to parallelize loops. A user finds his most time consuming loops in his code, and split them up between threads. In the following, we will give some simple examples to demonstrate the major features of OpenMP.

When parallelizing a loop in OpenMP, we may also use the schedule clause to perform different scheduling policies to effect how loop iterations are mapped onto threads. There are four scheduling policies available in OpenMP. The *static* scheduling method deal-out blocks of iterations of size "chunk" to each thread. In the *dynamic* scheduling method, each thread grabs "chunk" iterations off a queue until all iterations have been handled. In the *guided* scheduling policy, threads dynamically grab blocks of iterations. The size of the block starts large and shrinks down to size "chunk" as the calculation proceeds. This will achieve load balance among processors, especially only a few processors are used as shown below. Finally, in the *runtime* scheduling method, schedule and chunk size are taken from the OMP_SCHEDULE environment variable and hence are determined at runtime. These are only a few parallelization schemes available in OpenMP and will be used in our study. We will use these different schemes to measure the execution times and to find a best strategy to our code.

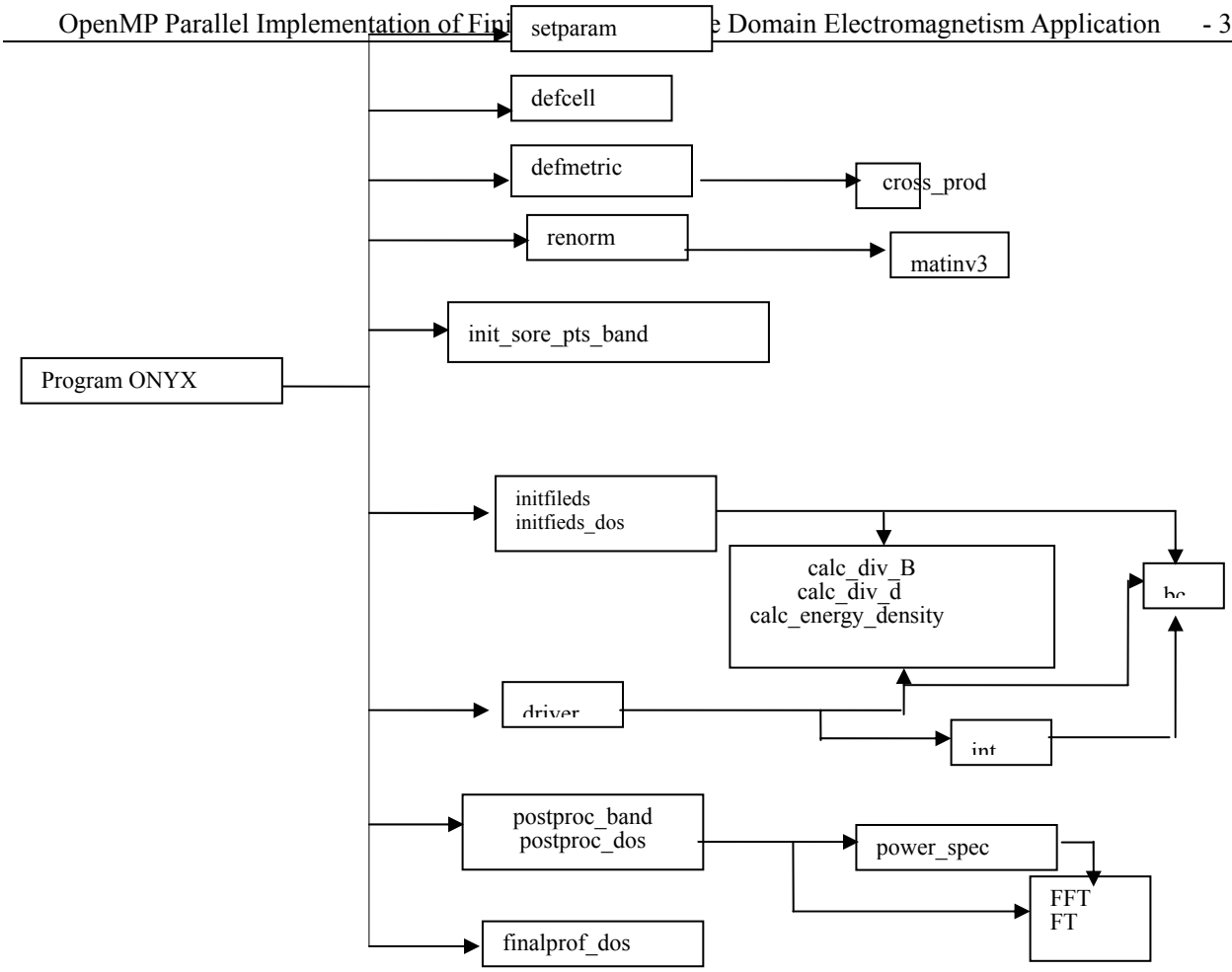


Figure1 Structure of the Program

4. PARALLEL IMPLEMENTATION

The Order N method (Finite Difference Time Domain) [Chan 1995] electromagnetism program demands large amount of memory space and is very time consuming. Hence, it is difficult to run the code on a single processor machine. The
main objective of this research project is to use OpenMP to implement the parallelization of this program so that larger applications can be simulated on parallel computers and faster
execution time is obtained.

The parallelized program is a Fortran 90 program with some OpenMP directives. At the moment the code can perform two basic types of calculation. The first is a band structure calculation, which starts off with an arbitrary set of initial fields corresponding to some wavevector K, integrates in time storing the fields at some random sampling points and then Fourier transforms into the frequency domain. The peaks in the frequency spectrum correspond to the frequency the allowed modes. The second type of calculation finds the time dependent Green's function by setting the initial fields to be a delta function in space for one of the field components, zero for all the others. The fields are integrated in time and stored at each time step to give *g(t, r, r')*. Fourier transforming gives *G(w, r, r')*.

This program is composed of 23 subroutines. Figure1 shows the structure of this program.

The main idea of our parallelization process is to find the most-time consuming subroutines and then use OpenMP to parallelize them incrementally. After compiling the code on our SGI Origin 2000 machine [Laudon 1997], we run the code under **ssrun** to generate profiling data and then use **prof** to translate the raw profiling data into a readable timing report. We get the following PC sampling file as shown in Figure 2, where **cum.%** means the percentage of the execution time that the functions consumed until now, and **samples** indicate the number of samples detected during the execution.

Total time=268.53 (secs)

Function	Secs	%	Cum.%	Samples
int	187.370	57.3%	57.3%	18737
bc_ymin_bloch	41.670	12.7%	70.0%	4167
bc_ymax_bloch	38.720	11.8%	81.9%	3872
fft	26.270	8.0%	89.9%	2627
bc_zmin_bloch	6.030	1.8%	91.7%	603
bc_xmin_bloch	5.970	1.8%	93.6%	597
bc_xmax_bloch	5.760	1.8%	95.3%	576
bc_zmax_bloch	5.550	1.7%	97.0%	555
power_spec	4.890	1.5%	98.5%	489
inifields	4.090	1.3%	99.8%	409
calc_div_d	0.220	0.1%	99.8%	22
calc_energy_density	0.16	0.0%	99.9%	16

Figure 2 Execution Times in Subroutines

It's obvious from Figure 2 that subroutine **int** cost the most time, it consumes 57.3% of the total execution time. And subroutine **bc_ymax_bloch**, **bc_ymin_bloch**, and **fft**

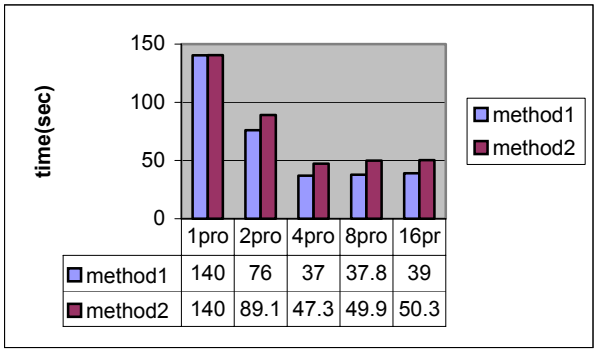
compose 11.8% , 12.7% , 8.0% of the total execution time, respectively. So our concentration focus on these subroutines Subroutine **int** forms the heart of the finite difference time domain calculation . It advances the electric and magnetic fields forward in time by *nt* time-steps . There are a few points that should be discussed here. Two sets of fields are stored e_cur and h_cur, which refer to the fields at the current time step, and e_prev and h_prev, which refer to the previous time step. The first thing the routine does at each time step is to switch round the current and previous fields so that current fields become the previous ones and vice verse. Some loops in the procedure have input/output and data dependence and cannot be parallelized. There are also three nested loops in the procedure **int**, which consume most of the processor time and have limited data dependence. Hence, we concentrate on the loops. It's a 3-layer nested loops as shown below.

```
do iz=1,izmax
do iy=1,iymax
do ix=1,ixmax
! Define Curl H
 curl(1)=h_prev(3,ix,iy,iz)-h_prev(3,ix,iy-1,iz) &
   &            -h_prev(2,ix,iy,iz)+h_prev(2,ix,iy,iz-1)
 curl(2)=h_prev(1,ix,iy,iz)-h_prev(1,ix,iy,iz-1) &
   &            -h_prev(3,ix,iy,iz)+h_prev(3,ix-1,iy,iz)
 curl(3)=h_prev(2,ix,iy,iz)-h_prev(2,ix-1,iy,iz) &
   &            -h_prev(1,ix,iy,iz)+h_prev(1,ix,iy-1,iz)

! Integrate fields in time

 e_cur(1,ix,iy,iz)=(1.0-sigma(ix,iy,iz))*e_prev(1,ix,iy,iz)
+  &
  &       (eps_inv(1,1,ix,iy,iz)*curl(1)   &
  &       +eps_inv(1,2,ix,iy,iz)*curl(2)   &
  &       +eps_inv(1,3,ix,iy,iz)*curl(3))

 e_cur(2,ix,iy,iz)=(1.0-sigma(ix,iy,iz))*e_prev(2,ix,iy,iz)
+  &
  &       (eps_inv(2,1,ix,iy,iz)*curl(1)   &
  &       +eps_inv(2,2,ix,iy,iz)*curl(2)   &
  &       +eps_inv(2,3,ix,iy,iz)*curl(3))

 e_cur(3,ix,iy,iz)=(1.0-sigma(ix,iy,iz))*e_prev(3,ix,iy,iz)
+  &
  &       (eps_inv(3,1,ix,iy,iz)*curl(1)   &
  &       +eps_inv(3,2,ix,iy,iz)*curl(2)   &
  &       +eps_inv(3,3,ix,iy,iz)*curl(3))
  enddo
  enddo
  enddo
```

Since OpenMP can only parallelize one loop, a limited parallelism can be exploited in the code if the original loops are used directly. In other words, we have no way to parallelize all the original nested loops. After careful study, we found that we can change the nested loop into one-layer loop manually. Since we know before hand that iymax=1, and izmax=ixmax=7, we change the loop into the following single loop manually:

```
!$omp parallel do
!$omp& default(shared)
!$omp& private(i,iz,iy,ix)
!$omp& schedule(guided,150)
do i=0, 48
ix=(i-(i/7)*7+1);
iz=((i/7)+1);
iy=1;
<loop body>
enddo
!$omp end parallel do
```

The above technique is called loop collapsing [Wolf 96]. Due to loop collapsing, we have to calculate indices in the new loop as shown in the code above. Now, the nested loops become a loop with longer loop limits and OpenMP can parallelize the whole loop easily. As also shown in the above program, we insert several OpenMP directives direct the parallelization process. The directive **!$omp parallel do** tells the compiler that we want to parallelize the next loop. Directive **!$omp& default(shared)** indicates to the compiler that all the variables are shared except the variables specified in **!$omp& private(i,iz,iy,ix)**. Finally, directive **!$omp& schedule(guided,150)** specifies the loop scheduling policy and chunk size. In the above example, we use guided scheduling policy and a chunk size of 150. As shown below, we have experimented various scheduling policies and chunk sizes to achieve better performance. Before the loop transformation, at most 7 concurrent threads can be generated. After the transformation, 49 parallel threads can be run in parallel, thus yielding much better performance. As shown in Figure 3, for this main loop, the method with loop collapsing (method 1) not only results in correct parallel program but also gives more parallelism than the original code (method 2).



	1pro	2pro	4pro	8pro	16pr
method1	140	76	37	37.8	39
method2	140	89.1	47.3	49.9	50.3

Figure 3 Performance Comparison after Loop Collapsin

5. PERFORMANCE ANALYSIS

The following graph (Figure 4) shows the relationship between speedup and the number of processors. We see a plateau in the speedup after the number of processors increased beyond 8. This is due to communication overhead caused in the parallelization process and data dependence within the original program. However, we expect better scalability will be achieved if the programm with larger array sizes are used in the future.

Different schedule strategies including static, dynamic, guided policies are used in our experiments to get the best policy and chunk size. Figures 5-8 show the execution times under different conditions with different number of processors.

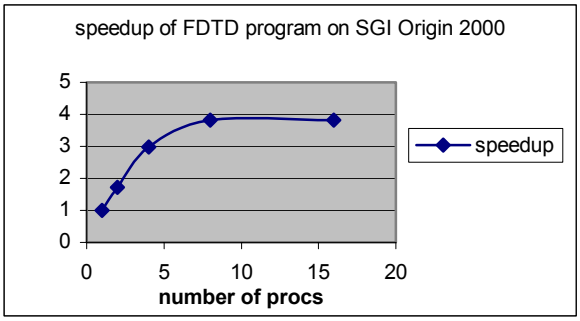


Figure 4 Speeding with Different Number of Processors.

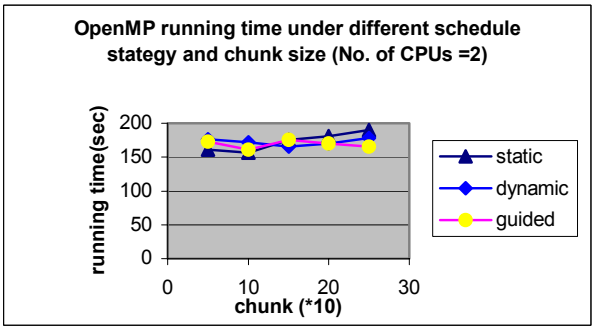


Figure 5 Running Times using Different Scheduling Policies

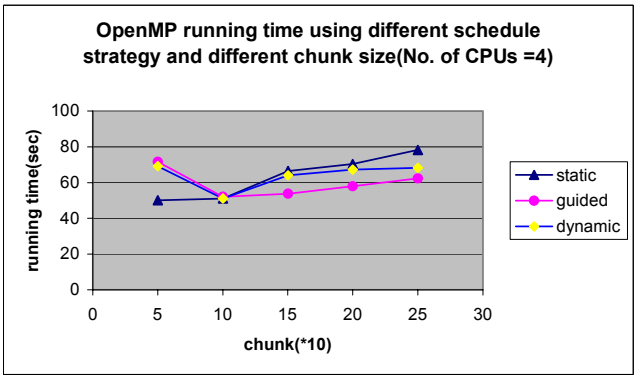


Figure 6 Running Times using Different Scheduling Policies And Chunk Sizes with 4 Processors

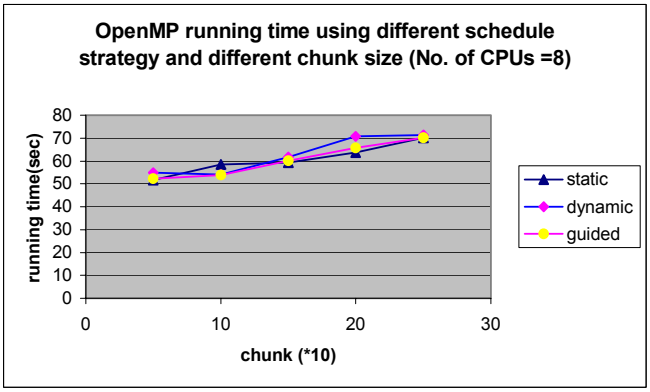


Figure 7 Running Times using Different Scheduling Policies and Chunk Sizes with 8 Processors.

We infer from these graphs that when the number of threads is small, static schedule is more efficient than guided, and guided schedules are better than dynamic ones. But when the number of threads increases to more than 8, guided schedule strategy is more efficient than static schedule, and static schedule is better than dynamic schedule. Table 1-3 show the profiling details of FDTD program executing on SGI Origin 2000 when the number of processors is 2, 4, and 8, respectively.

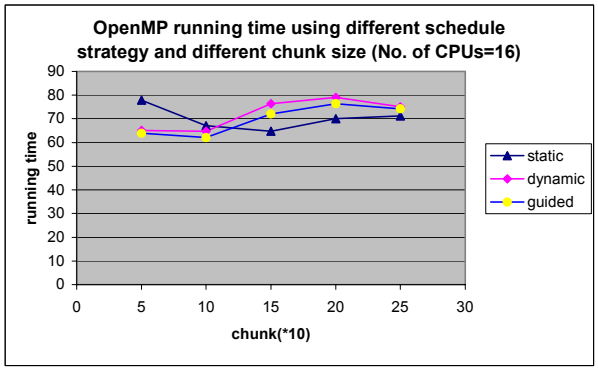


Figure 8 Running Times using Different Scheduling Policies and Chunk Sizes with 16 Processors.

Table 1 The number of CPUs = 2, total time= 139.52 (secs)

Function	Secs	%	Cum.%	Samples
_mpdo_int_2	41.68	27.1%	27.1%	5168
_mpdo_int_2	30.590	212.09%	48.2%	3959
bc_ymax_bloch	20.48	14.12%	62.32%	1048
bc_ymin_bloch	18.760	12.9%	75.2%	876
fft	14.50	10.0%	85.2%	650
int	3.470	2.39%	87.59%	347
_mpdo_bc_ymax_bloch_1	2.540	1.75%	89.34%	254
_mpdo_bc_zmax_bloch_1	2.330	1.6%	90.91%	233
_mpdo_bc_ymin_bloch_1	2.110	1.455%	92.36%	211
_mpdo_bc_zmin_bloch_1	2.100	1.45%	93.81%	210
power_spec	1.330	0.917%	94.76%	133
bc_xmax_bloch	0.950	0.655%	95.41%	95

Table 2 The number of CPUs = 4, total time=63.38 (secs)

Function	Secs	%	Cum.%	Samples
_mpdo_int_2	16.230	25.6%	25.6%	1623
_mpdo_int_1	10.230	16.1%	41.7%	1023
fft	5.410	8.5%	50.3%	541
bc_ymax_bloch	5.120	8.1%	58.4%	512
bc_ymin_bloch	5.06	8.0%	66.6%	506
int	4.530	7.1%	73.5%	453
_mpdo_bc_zmin_bloch_1	3.630	5.7%	79.2%	363
_mpdo_bc_zmax_bloch_1	2.850	4.5%	83.7%	285
_mpdo_bc_ymax_bloch_1	2.730	4.3%	88.0%	273

_mpdo_bc_ymin_bloch_1	2.700	4.3%	92.3%	270
power_spec	0.98l	1.5%	93.8%	98
_mpdo_bc_xmax_bloch_1	0.73	1.2%	95.0%	73
bc_zmin_bloch	0.69	1.1%	96.1%	69
bc_zmax_bloch	0.67	1.1%	97.1%	67

Table 3 The number of CPUs = 8, total time=55.40 (secs)

Function	Secs	%	Cum.%	Samples
_mpdo_int_2	13.960	25.2%	25.2%	1396
fft	8.460	15.3%	40.5%	846
_mpdo_int_1	6.470	11.7%	52.1%	647
int	5.050	9.1%	61.3%	505
bc_ymax_bloch	4.800	8.7%	69.9%	480
bc_ymin_bloch	4.590	8.3%	78.2%	459
_mpdo_bc_ymax_bloch_1	2.430	4.4%	82.6%	243
_mpdo_bc_ymin_bloch_1	2.010	3.6%	86.2%	201
power_spec	1.320	2.4%	88.6%	132
_mpdo_bc_zmin_bloch_1	1.270	2.3%	90.9%	127
_mpdo_bc_zmax_bloch_1	1.080	1.9%	92.9%	108
_mpdo_bc_xmax_bloch_1	0.700	1.3%	94.1%	70
inifields	0.67	1.2%	95.3%	67

6. CONCLUSION

The FDTD method applied to a discretized form of Maxwell's equation has been parallelized using OpenMP. The FDTD method provides an accurate simulation of photonic waveguide devices that are required for future communications technology. We rapidly implemented the parallelization using OpenMP and compared the run time against the serial version of the program. We find the computation speeds up by about four times using 8 processors. Of course, when the problem size changes, the scalability and efficiency may also change. Theoretically, when the problem size is bigger, the ratio of computation over communication is also getting bigger. This implies that the communication overhead becomes relatively smaller. It is our conjecture that the OpenMP parallel code will be more scalable and efficient when computing larger problems. Our program is also applicable to a wide range of problems including nonlinear effects and dispersion. It can be applied to pulsed or continuous wave propagation.

7. REFERENCE

[1] R. Allen and K. Kennedy, Optimizing Compilers for Modern Architectures, Morgan Kaufmann Publishers, 2002.

[2] C. T. Chan and Q. L. Yu and K. M. Ho, Order-N Spectral Method for Electromagnetic Waves, Physical Review B, 51, 23, pp. 16635, 1995.

[3] R. Chandra, R. Menon, L. Dagum, D. Kohr, D. Maydan, and J. McDonald, Parallel Programming in OpenMP, Morgan Kaufmann Publishers, October 2000.

[4] J.W. Haus, "A brief review of theoretical results for photonic band structures," J. of Modern Optics 41, 195, 1994.

[5] N. Grote and H. Venghaus (Eds.), Fibre Optic Communication Devices, Springer Verlag, Berlin, 2001.

[6] J. Laudon and D. Lenoski, The SGI Origin: A ccNUMA Highly Scalable Server, the 1997 International Symposium on Computer Architectures, Denver, Co, pp. 241-251.

[7] A. Taflove, Computational Electrodynamics: the finite difference time-domain method, Artech House, Boston, 1995.

[8] A.J. Ward, Transfer Matrices, Photonic Bands and Related Quantities, PhD Dissertation,University of London, 1996.

[9] A.J. Ward and P.B. Pendry, "A program for calculating photonic band structures and Green's functions using a non-orthogonal FDTD method," Computer Physics Communications 112, 23, 1998.

[10] Weisbuch, H. Benisty, S. Oliver, M. Rattier, C.J.M. Snith and T.F. Krauss, "Advances in Photonic Crystals," Physica Status Solidi B 221, 93, 2000.

[11] M. Wolfe, High Performance Compilers for Parallel Computing, Addison-Wesley Publishing Company, 1996.

A New Method of Image De-noise Based on Wavelet Packet And Median Filtering*

Wei Chen
School of Information Technology, Wuhan University of Technology
Wuhan, Hubei, 430063, P.R. China ,
E-mail: chenlin@public.wh.hb.cn
And
Xiaoming Ren, Weixia Liu
School of Navigation, Wuhan University of Technology
Wuhan, Hubei, 430063, P.R. China,
E-mail: rxm_123@163.net

ABSTRACT

Gauss color noise can be filtered, via the threshold operation on wavelet packet transform of images; however, there is usually much impulse noise in the images. While applying median filtering will effectively reduce the impulse noise, combining wavelet packet transform and median filter will also be an effective approach for removing the noises.

Keywords: Wavelet Packet Transform, Threshold Operation, And Median Filtering, impulse noise, Gauss color noise

1. INTRODUCTION

Wavelet transform has achieved good applied effects in many fields as a brand-new mathematical method of analyses and image processing is one of the first applied and most mature fields. Image de-noise is widely used as a technology of image processing, which can raise the image signal-to-noise ratio and highlight image expected features. According to its sources, image noise is divided into additive noise; multiply noise, quantification noise, salt and pepper noise and so on. Yet, according to its property, they can be named Gauss color noise and impulse noise. In 1974 Tukey became the first person to apply nonlinear median filtering in image processing. His method is increasingly popular in the fields of image processing, because it could effectively reduce impulse noise when protecting the details of images. Mean filtering takes the mean value of domains (windows) of image element for the output value to effectively diminish Gauss noise with zero mean value. However, the linear method also destroys the high-frequency signals (such as edge and details). Lee and Kassam put forward an improved modified method, trimmed mean filtering (MTM), which combined the running mean and median filtering with the aim of reducing Gauss noise and impulse noise simultaneously.

The operating procedures of MTM filtering are as follows

1. While handling the listed-k picture element, first select the gray scale median m_k in the filtering windows;
2. Taking m_k for the center, then select a gray scale interval $[m_{k-q}, m_{k+q}]$;
3. Average all the points in the interval, taking care of the results as the final filtering to output.

The mathematical formulae are as follows:

$$y_k = \text{average}\{x_i, m_{k-q} \le x_i \le m_{k+q} \text{and} i \in w_k\} \qquad (1)$$

W_k stands for the image formed by the picture elements in the filtering windows. The fact is that the filtering effects are not very ideal by MTM, because the q can directly affect the results of filtering and is protecting the ability to image details. That is, the smaller the q is, the closer MTM is to the median filtering and the stronger the protecting ability is, meanwhile, the function of jamming Gauss noise diminishes. Contrarily, the bigger the q, the closer MTM is to the mean filtering and Gauss noise is jammed effectively while the images turn into blurring. This article purposefully adopts wavelet packet transform for the threshold operation and also median filtering to reduce Gauss noise and impulse noise.

2. DECOMPOSITION AND ALGORITHM OF WAVELET PACKET

To explain the decomposition of the wavelet packet，first we need know about the conception of wavelet packet. Suppose $\{V_k\}$ is a multi-rate resolving space sequence in $L^2(R)$ and $\{W_{k+1}\}$ is the orthogonal complement space relating V_{k+1} to V_k. Again, suppose $\phi(x)$ and $\psi(x)$ are homologous orthogonal scale and wavelet function. Then, the following are the double scale equations:

$$\begin{cases} \phi(x) = \sum_k h_k \phi(2x-k) \\ \psi(x) = \sum_k g_k \phi(2x-k) \end{cases} \qquad (2)$$

$\{h_k\}$ and $\{g_k\}$ are conjugate electric separator and $g_k=(-1)h_{1-k}$.

if $\mu_0(x)=\phi(x)$， $\mu_1(x)=\psi(x)$

$$\begin{cases} \mu_{2n}(x) = \sum_k h_k \mu_n(2x-k) \\ \mu_{2n+1}(x) = \sum_k g_k \mu_n(2x-k) \end{cases} \qquad (3)$$

The function family $\{u_n(x) \mid n \in z\}$ is an orthogonal wavelet packet versus orthogonal scale function $\phi(x)$. Function family $\{2^{j/2} u_n(2^j x-k) \mid k, j \in z, n \in z\}$ derived from $\phi(x)$ are called normal orthogonal basis.When j is fixed, $\{2^{j/2} u_n(2^j x-k) \mid k, j \in z, n \in z\}$ forms a norm orthogonal basis in $L^2(R)$. When n is fixed, $\{2^{j/2} u_n(2^j x-k) \mid k, j \in z\}$ forms a norm orthogonal basis in $L^2(R)$.

* This paper is supported by Hubei Province Natural Science Funds.

<table>
<tr><td>LL3</td><td>Hl3</td><td rowspan="2">HL2</td><td rowspan="3">HL1</td></tr>
<tr><td>LH3</td><td>HH3</td></tr>
<tr><td>LH2</td><td>HH2</td><td>HH2</td></tr>
<tr><td colspan="2">LH1</td><td colspan="2">HH1</td></tr>
</table>

Figure 1 A Triple Cascade of Images Caused by The Wavelet Transform

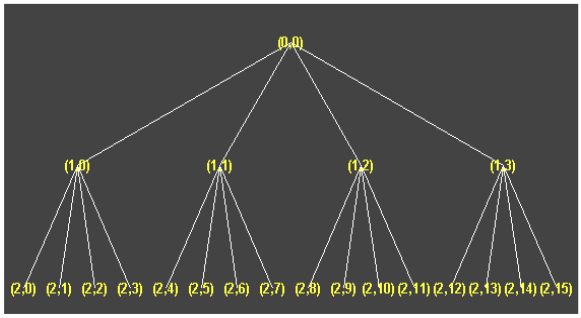


Figure 2 Wavelets Packet Decomposition

To be simple, assume

$$U_j^n = \text{clos}\left\{2^{j/2}u_n(2^j x-k) \mid k, j \in z, n \in z\right\} \quad (4)$$

In particular, ， n=0，

$$U_j^0 = \text{clos}\left\{2^{j/2}u_n(2^j x-k) \mid k \in z\right\} = V_j \quad (5)$$

n=1，

$$U_j^1 = \text{clos}\left\{2^{j/2}u_n(2^j x-k) \mid k \in z\right\} = W_j \quad (6)$$

Obviously，in the light of wavelet packet theory, we know

$$V_j = V_{j+1} \oplus W_{j+1} \quad , \quad \forall j \in z \quad (7)$$

(3) – (5), then

$$U_j^0 = U_{j+1}^0 \oplus U_{j+1}^1 \quad , \quad \forall j \in z \quad (8)$$

It can be proven that this conclusion is suitable for other natural numbers, namely, ∀n∈z; there is

$$U_j^n = U_{j+1}^{2n} \oplus U_{j+1}^{2n+1} \quad , \quad \forall j \in z \quad (9)$$

Therefore, $L^2(R) = \bigoplus_{j \in z} W_j, \forall j \in z$ (10)

Eq. (9) indicates that by the decomposition of a wavelet packet with multi-rate resolution analyzing any subspace V_n， one may obtain its binary tree decomposition forms.

The wavelet transform is often used in two-dimensional image processing and also through horizontal and vertical filtering, the discrete wavelet transform divides primordial image into four parts: vertical and horizontal-orientation low-frequency sub-tape LL1, horizontal-orientation low- frequency and vertical-orientation high-frequency sub-tape LH1, horizontal-orientation high-frequency and vertical-orientation high-frequency sub-tape HL1, vertical and horizontal-orientation high-frequency sub-tape HH1. Afterwards, we can further obtain the low-frequency sub-tape LL1 and get the additional four sub-tapes LL2, LH2, HL2, HH2 with lower resolutions. Applying again, we can realize the multistage decomposition on images. As figure 1 shows, it is a triple cascade of images. While, besides re-decomposing low-frequency sub-tapes on the basis of first layer decomposition, wavelet packet still makes further decompositions on other three high-frequency sub-tapes. From figure 2. We see that a wavelet packet can preserve the primordial information more effectively.

The decomposition algorithm of the wavelet packet is as follows:

$$\begin{cases} d_l^{j+1,2n} = \sum_k h_{k-2l} d_k^{j,n} \\ d_l^{j+1,2n+1} = \sum_k g_{k-2l} d_k^{j,n} \end{cases} \quad (11)$$

$d_l^{j+1,2n}$、$d_l^{j+1,2n+1}$、$d_k^{j,n}$ separately are the coefficients of subspace U_{j+1}^{2n}、U_{j+1}^{2n+1}、U_j^n，，h_k is the low pass electric separator coefficient, $g_k = (-1)^k h_{1-k}$.

Reconstructing the algorithm of wavelet packet:

$$d_l^{j,n} = \sum_k [h_{l-2k} d_k^{j+1,2n} + g_{l-2n} d_k^{j+1,2n+1}] \quad (12)$$

3. MEDIAN FILTERING

Median filtering, based on the theory of ordering statistics, is a nonlinear signal processing technique, which can effectively jam noise. Median filtering can make a reduction to the noise (especially impulse noise) and at the same time it most likely protects detailed information (for example, edge, sharp angle). The principles of median filtering are very simple. Scan the images with a window W, and then enlist the picture elements comprised in the window according to the rising or sag order of the degree of gray scale. If you fetch the middle one in the gray scale values as the gray scale of the window center picture elements, the course for median filtering will be finished. Formula is the following one:

$$g(m,n) = median\{f(m-k,n-l),(k,l) \in W\} \quad (13)$$

As usual, the number of picture elements in windows is odd in order to have at least a medial picture element. If that is an even number，we can figure out the average of the two gray scale of medial picture elements。

4. THRESHOLD METHOD OF WAVELET PACKET

Let us suppose we have obtained the observed data

$$y_i = x_i + \sigma n_i \quad (i = 1,2...N) \quad (14)$$

n_i is Gauss color noise with zero mean, σ is its variance, x_i is the expected signal, y_i is the observed value. We can see that how to precede filtration reduces to the problem of how to restore the expected value x_i from the observed values y_i.

The reasons for applying wavelet packet transform to jam noise mainly lie in the better expressed local features of structures, and from those features the displayed discrepancies which are different from those of noise as well. Yet the discrepancy is the main basis for distinguishing signal and noise. The wavelet transform of Gauss noise is still a Gaussian distribution, evenly distributed in each part of phase space. However, due to the signal limitation, coefficients of wavelet packet transform merely focus on a small-scale phase space. Hence from the energy point of view，in the wavelet domain, all coefficients of the wavelet packet transform make a contribution to the noise- i.e. The energy of noise is mainly found from the coefficients of the wavelet, while only a

fraction of them are dedicated to the signals.

Accordingly, wavelet coefficients fall into two kinds: one acquired after the noise transform, whose numerical values are small; the other transformed from signals, including the results of noise changing, where the numerical values are large. Therefore, we can, via the discrepancy of the numerical values of the wavelet coefficients, fabricate a new method for reducing noise, namely, set up a threshold. All the wavelet coefficients greater than the threshold can be derived from the signal transform and can be preserved; while those less than the threshold can be thought of as being derived from noise, and eliminated.

To sum up, the process of wavelet packet de-noise could be described by the following three steps：

1) Observed values undergo wavelet transforms and also get the wavelet coefficient $\gamma = wy$.

2) In the wavelet domain, carry out a threshold operation on the coefficient γ of the wavelet packet. Two methods may be used:

Hard threshold operation,

$$\hat{x} = T_h(\gamma, t) = \begin{cases} \gamma & |\gamma| \geq t \\ 0 & |\gamma| < t \end{cases} \quad (15)$$

Soft threshold operation,

$$\hat{x} = T_h(\gamma, t) = \begin{cases} \mathrm{sgn}(\gamma)(|\gamma| - t) & |\gamma| \geq t \\ 0 & |\gamma| < t \end{cases} \quad (16)$$

Where t is the threshold, and $\mathrm{sgn}(\gamma)$ is γ symbol.

3) Wavelet packet anti-transform $\hat{x} = M\hat{x}$

then get the estimated $\hat{x}$ of x which is also the result information of de-noise.

5. CONCLUSIONS & ANALYSIS

During the experiment, we intentionally added Gauss noise into the primordial image, and then adopted the new method of wavelet packet threshold for de-noise; we also adopted a Symlet wavelet function for wavelet packet transform.

The image size is 256 × 256. Two threshold methods were applied in the operation. First, conduct wavelet packet hard threshold operation on images as a whole, and then the soft threshold operation is used on the images. Applying Stanford unbiased risk reckoning, the threshold is $t = \sqrt{2\ln(n\log_2(n))}$. The experiment results are shown in figure 3.

Judging from the results, the new de-noise methods of wavelet packet and median filtering are indeed fruitful in the practical processing of images. From the two points of signal-to-noise ratio (SNR) and peak signal-to-noise ratio (PSNR) to balance indexes, both indexes of images after their noise were reduced, then greatly raised, separately SNR=22.0611 and PSNR=44.3944. The formulae for SNR and PSNR are as follows:

$$\begin{cases} SNR = 10\log_{10}\dfrac{\sigma^2}{D} \\ PSNR = 10\log_{10}\dfrac{255^2}{D} \end{cases} \quad (17)$$

$$D = \frac{1}{NxM}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}|x(i,j) - \hat{X}(i,j)|$$

$$\sigma^2 = \frac{1}{N}\sum_{i=0}^{N-1}(x_i - \bar{x})^2$$

$$\bar{x} = \frac{1}{N}\sum_{i=0}^{N-1}x_i$$

The new method of applying the combination of Symlet wavelet packet transform and median filtering for de-noise, including Gauss noise and impulse noise at the same time, will clearly be able to help us obtain better results, when compared with simply using Symlet wavelet packet transform or using median filtering.

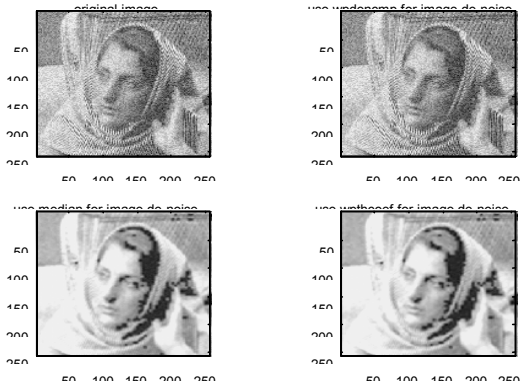


Figure 3 Results of De-noise for Wavelets Packet Transform and Median Filtering

6. REFERENCES

[1] Romberg, JK; Choi, H; Baraniuk, RG. Bayesian Tree-Structured Image Modeling Using Wavelet-Domain Hidden Markov Models. IEEE Transactions on Image Processing. [IEEE Trans Image Process], Vol. 10, No.7, pp1056-1068, July 2001.

[2] Tu Dan, Shen Jianjun, Shen Zhenkang. Applied Studies on Wavelet Threshold Tech in the Image De-noise. Journal of National Defense Science and Technology University. No. 2. Feb. 1999.

[3] Wang Yanhui, Wang yongchen. Wavelet Transform Technology on Image Signal De-noise. Journal of Shen Yang Engineering Institute. No.4. Dec.2000.

[4] Zhao Ruizhen, Qu Hanzhang, Song Guoxiang. Algorithm of Threshold Filtering Based on The Regional Correlation of Wavelet Analyses. Journal of Xi'An Electron Science and Technology University. No. 4. June 2001.

[5] Brown, RK. Image Registration Using Redundant Wavelet Transforms. Performer: Air Force Inst. Of Tech., Wright-Patterson AFB, OH. School of Engineering and Management. Mar. 2001.90p. Report: AFIT/ GE / ENG / 01 M-21.

[6] Lee, YH; Kassam, SA. Generalized Median Filtering and Related Nonlinear Filtering Techniques. IEEE Trans. Acoust. Speech Signal Proc., Vol. Assp-33, No. 3, pp. 672-683, 1985.

[7] Cheng Zhengxing. Wavelet Analysis Algorithm and Application. Xi'An Transportation University Publisher. May 1998.

Java Based Distributed Virtual Reality Construction

Yan Xinqing, Zhu Lijuan, Li Wenfeng
Department of Logistics Enginerring, Wuhan University of Technology
Wuhan, 430063,PRC
E-mail: yxq375@sohu.com
Chen Dingfang
Laboratory of Intelligent Information Processing, Academic of Science,
Beijing, 100080,China
E-mail: dfchen@public.wh.hb.cn

ABSTRACT

In this paper, we proposed constructing distributed VR system using Java. We discussed constructing Virtual Environment with Java 3D and accessing the data of the object in a relation database. We suggest using socket, RMI or XML as the communication tools for the communication among systems.

Keywords: Distributed Virtual Reality, Virtual Environment, Java 3D, RMI.

1. INTRODUCTION

Distributed Virtual Reality (DVR) nowadays is widely used in a lot of fields such as manufacture simulation, military command control system and even in computer games. Ideally a DVR software can be run on a variety of platforms, such as Linux, Unix or Microsoft Windows with little or even no modification, which brings a lot of pressure and problems to the software designer.

A DVR software can be divided into two parts: The painting part and the communication part. The painting part is responsible for the construction of the 3D virtual environment (VE), and displays the objects of the virtual world and their movements; the communication part takes control of communication among VEs, it send message back and forth through network among the VEs. The painting part modifies the VE according to the collaborative message received from the communication part and sends the message about its state to other VEs.

Many features of Java make it a suitable programming language for DVR. First, Java is an object-oriented language, which allows us to model the virtual object in the VEs or the DVR system easily. Second, Java is robust and net-based, software written in Java can be run under distributed computer system reliably. Third, Java is an architecture-neutral and portable programming language, application written with Java, without modification, can be run under a lot of platforms ranged from mainframe to portable PC, and the applet written Java can be run in the WEB explorer environment

Besides, many packages extend the programming ability of Java. Java 3D, a package provided by Sun, makes it convenient for Java to deal with 3D computer graphics and provides a way to construct virtual reality systems.

2. CONSTRUCTION OF VE WITH JAVA3D

Based on different low-level implementations, Java 3D has two versions now: one is based on OpenGL, another is based on Microsoft DirectX. Both versions uses JN I(Java Native code Interface) to accelerate the software running speed. Java 3D packages the detail implementation into classes so that we cannot feel the difference between these two versions. Usually we use the OpenGL implementation. We can download the Java 3D SDK and Runtime Environment from www.java.sun.com.

Java 3D provides hundreds of classes to support 3D programming, these classes include 3D Object, 3D transformation, 3D shade model, 3D texture, fog, 3D stereo sound and etc. Through these class, we can construct the Virtual Reality software easily.

Virtual Environment construction is an important aspect of Virtual Reality because that through the VE we get the feeling of reality. In Java 3D we construct the 3D Virtual Environment in a scene graph.

Scene Graph of Java 3D

Java 3D program is based on Scene Graph, which is consisted of many Java 3D objects called Nodes. All nodes are arranged in a tree structure as follows:

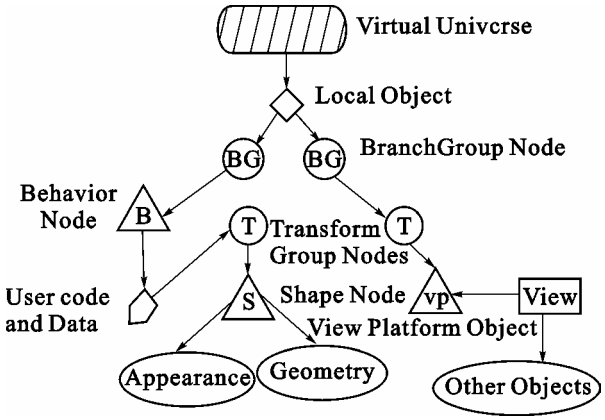


Fig.1 Scene Graph of Java 3D

We can classify the Nodes into two catalogs: the Group nodes and the Leaf nodes. The Group node can contain one or more sub-nodes but can only have one parent node. The Leaf node object can be of geometry shape, light model, fog, LOD, 3D stereo sound etc and can be shared among Group Nodes.

In the program using Java 3D, we must define a Universe object and a Local Object first to hold the Scene Graph (Although we can define many Universe and Locale, we usually define only one). And then we define one or many BranchGroups. In a BranchGroup we can define some objects or a Transform Group node to represent the movements of the objects.

To display an object in the VE, we need a Geometry object to represent its shape and an Appearance object to represent the appearance (such as color, texture etc), In order to give the feeling of reality, we often construct the Appearance by using Texture.

Object definition:

Java 3D only provides some basic object definitions directly,

these basic objects include facet, sphere, box, cylinder, cone and etc. For the complex objects in the VE, although Java 3D doesn't provide definition for them directly, we can combine some simple 3D objects together to construct them.
Another way for us to construct complex objects is through QuadArray or TriangleArray. Usually we can divide the complex object into a lot of Triangles, which form a TriangleArray. Java 3D provides support for the TriangleArray, and we can use it directly. For example, we can define a piece of irregular ground into triangle array and by painting the triangles independently we construct the ground.
The third way to construct complex objects is to model them in CAD Software and export the models to VRML files. Java 3D reads the data model from the VRML file and paints the object respectively.
Because there may be too much objects in the VE for a computer to process and display efficiently, we can define a bound, only objects in the bound can be processed and displayed by the computer, objects out of the bound will not. In java 3D, we can define three kinds of bounds: the BoundingSphere, the BoundingBox, and the BoundingPolytop,
Besides, we can define some objects as LOD, the shape of the LOD object may change according the distance between the viewpoint and the object.
To give the feeling of reality, 2D image texture can be used to construct the appearance of the object in the VE, and we can also combine the light model, the fog effect and the 3D stereo sound to make the VE more reality.

Movement representation
In a VE, there may be a lot of movements of the objects. In Java 3D, we put the moving object into different BranchGroup and use the Alpha object and Interpolator interface of Java 3D to construct the movement.
The Alpha object defines the aspects related to the speed and time of the movement, such as movement repetition times, time for each repetition and etc. It divides a movement repetition into five periods: starting period, uprising period, up period, descending period and down period, we can allocate the time for different period to change the movement of the object.
The Interpolator is an interface to control the way of the movement. Through its subclass we can make the object move according to a special path or way. Some of the main sub classes of Interpolator are:
PositionInterpolator class: let the object move smoothly between two 3D points according the way an Alpha object specifies.
RotaionInterpolator: let the object rotate between two angles, the rotation speed is specified by an Alpha object.
ScaleInterpolator: change the size of the object according the way an Alpha specifies.
SwitchValueInterpolator : change the object to different shapes at different time.
PositionPathInterpolator: let the object move in accordance with a specified path in the way an Alpha object defines.
RotPosPathInterpolator: the object rotates around an axis in the way an Alpha object defines while moving along a special path.
We can call the Enable method of the Interpolator to start the movement and Disable method to stop the movement.

Interaction between human and VE
Using 3D geometry shapes, the Alpha object and the Interpolator interface, we can construct the Virtual Environment with moving and static objects. But we still need a way to interact with the VE objects through mouse, keyboard or some other devices. Java 3D provides some classes for the interaction.
Keyboard: this class monitors the keyboard input, and let the object act according to the key we pressed. For example, when we press '+' or '-', we actually move the viewpoint forward and backward; when we press UP or DOWN, we make an object rotate around the Y-Axis. Java 3D provides many keypresses to let use interact with the software.
Mouse: there are three mouse interaction classes: MouseRotate, MouseZoom and MouseTranslate, which make the object rotate, zoom or translate according to a specified mode.
Picking Object: through the picking object in com.sun.j3d.utils.behavior, we can pick objects in the VE. It has several sub classes such as PickRotateBehavior, PickTranslateBehavior and PickZoomBehavior to let us pick objects more easily in the 3D virtual environment.
After we pick the object, we can change the movement, the shape and the appearance of the object through a method calling provided by Java 3D.

Constructing VE from a database
Generally, the complexity of the VE and the objects will produce a great deal of information to be stored. We need to store these information in a database, these data includes the shape, the location, the color, the texture and etc. of the VE and objects. Through JDBC, Java Database Connectivity, we can store these information in a database and access the data later.
JDBC is a standard driver for the connection between the Java program and database system. Through JDBC, we can access the data stored in database such as Oracle, IBM DBⅡ or some desktop database management system. For some DBMS without JDBC driver, such as SQL Server, we can use JDBC-ODBC-bridge to implement the connection.
Usually, we may connect to the database and fetch the information, construct the 3D objects accordingly in the initialization part of the software. And then we can display the objects and their movement later in the paint part

3. COMMUNICATION AMONG VES

Because Java is designed to be an Internet programming language, so it provides a lot of ways for the communication through computer network among programs. Usually we can use one or some of following methods to implement the communication.

Communication through Socket
The simplest way to communicate between two programs is through socket. To use socket for communication, we must define a socket first. And then through the socket, we can send message to other computers, and we can listen the socket to receive data sent by other computers.
When using a socket in DVR, there must be a message cycle in the DVR program. It listens to the socket, after having received the data sent by other computers, it responds according to the data. And through socket, it can also send data to other programs.
In a DVR software, the program will often redraw some of the objects and change the movement of the objects in the VE according to the data sent through a socket by other programs. After the reaction, it will usually produce its own message and send the message to other systems running in the network to let them know. In this way, VEs running in different

computers can coordinate their action and display.

Communication Using RMI

Another way for software written in Java to communicate with each other is through RMI (Remote Method Invocation). Through RMI, one program can invoke a method of a class running in a different computer in the network, thus using Java we can build disturbed system easily.

Although RMI programs can be divided into client side and server side, a program can be both. The server side bounds a name with the remote object in the RMIRegistry, The client side look up the RMIRegistry for the name, After getting the reference to the remote object, the client side invokes the RMI through a stub in the its side.

Using RMI, we can build the DVR system in the following way: Each VR system written in Java is running in a different computer connected within a network independently, but it will provide a RMI method to tell its current state to other programs. Another system will invoke this method and get its state data, and according the data, this program will react in some way and change its state.

Large scale communications using XML through Internet

Although socket and RMI provide the basic way for software to send and receive message to and from each other, for large scale Distributed Virtual Reality systems, especially running and communicating via the Internet, these two ways are difficult or even impossible to maintain. We need to use XML (extended Markup Language) to communicate with each other. XML is a standard communication language (proposed by W3C) for data exchange through the Internet. It can be sent via the Internet through HTTP and can pass through the enterprise firewall transparently, and now is widely used in B2B and B2C applications. When using XML as a communication language among applications written in Java, we need to use SAX or DOM to analysis the format and content of the XML file.

Now in large scale Internet based DVRs, we can use XML for the exchange of data among different VEs. Every VE constructs a XML document to represent the objects and their states and some coordination information, afterwards it sends this document to other applications through the Internet. Other applications receive the document and analysis the content, get the information and modify it's own VE scene accordingly. If a VE has some information to tell other VEs, it forms a XML file and sends it to other applications through the Internet.

4. CONCLUSION

In the project Distributed Virtual Design/ Virtual Manufacture sponsored by Science and Technology Officer of Wuhan, We are using Java 3D and other tools constructing a Distributed Virtual Reality system. But there are still some problems to be taken into deep consideration in the future:

- The autonomy of each VE: When communicating through the Internet, the transfer may be delayed or even the data may be lost. So each VE should run independently, and after receiving some information, it modify itself accordingly.,
- The store of the objects: Because of the complexity of the VE, its objects and their movement, it's difficult to store the all these information in a traditional relation database. There we need a new way to store them efficiently.
- The efficiency of the Java running environment: Because Java program runs on bytecode, although Java 3D uses JNI to accelerate the 3D drawing speed, but the result is still of a little unsatisfactory.

5. REFERENCES

[1] Bruce Eckel, Thinking in Java the second edition [M], www.bruceeckel.com, 2001

[2] Sun Corp. The Java 3D Api Specification version 1.2[M] www.java.sun.com, 2000

[3] Yang Baomin, Zhu Yining, Distributed Virtual Reality Technology an Application[M], Science Press, 2000

[4] Wang Shao-feng, Wang Ke-hong, The Design of RMI-Based Workflow Management System[J], Computer Integrated Information System, 2000.10: 58-63

[5] Yang Meng-zhou, Pan Zhi-geng, Shi Jiao-ying, Architecture of Distributed Virtual Reality[J], Research of Computer Application, 2000.7:2-4

Performing Firearm Identification Ballistic Database Operation Based on An Intranet

D.G. Li
Department of Computer Science, Edith Cowan University
2 Bradford Street, Mount Lawley, Western Australia 6050, Australia
E-mail: d.li@ecu.edu.au
And
C. Jiao
Department of Computer Science, Edith Cowan University
2 Bradford Street, Mount Lawley, Western Australia 6050, Australia
E-mail: cjiao@student.ecu.edu.au

ABSTRACT

Computerized desktop BFISs use modern technologies to improve its efficiency, correctness, and scalability from its traditional manual systems. To further develop an Intranet BFIS, an Intranet integrated DBMS is required. This is not only to provide Intranet assessment but also to improve the management of scale, preferment, security and image data.

The IBFIS uses MSS2000 as its DBMS and receives above benefits over the desktop BFIS. By using the reverse engineering technique provided by PowerDesigner, IBFIS DBMS starts its database with the contents of FireBall's data, collects further firearm ballistic information over the Intranet, provides searched result over the Intranet, and provides the opportunity to FireBall users to refresh their MSA Fireball database with the contents of IBFISDB.

The prospect of IBFIS is significant as matching a recovered cartridge to its proposed firearm is important for both solving the crime and providing forensic scientific evidence in the Court. Hence IBFIS increases the chances for law enforcement.

Keywords: Forensic Science, DBMS, Intranet, Reverse Engineering, Image Data

1. INTRODUCTION

Computerized Ballistic Firearm Identification System (BFIS) uses modern technologies to improve its efficiency, correctness, and scalability from its traditional manual identifying systems [1]. Such computerize system always use a complicated image-storage-capable database to support its applications. FireBall is one of computerized BFIS using Microsoft Access [2] (MSA) as its database. To further develop such BFIS into an advance Intranet BFIS (IBFIS), Microsoft SQL Server 2000 [3] (MSS2000) is used as a database management system (DBMS) to improve the level of accessibility, scalability, reliability and security.

In this paper we describe the tools and processes involved to develop IBFIS database (IBFISDB) base on Fireball database (FBDB). The project is aim to develop an integrated and secured DBMS to supports the IBFIS. It also provides an opportunity for desktop application Fireball users to use the MSS2000 database in its MSA form so that the large and rich ballistic database can be shared in different level of usages.

In this paper, after an introduction of BFIS, we will first review the history of FBDB and the development of IBFIS DBMS. Next, we will compare two representative and popular DBMSs, MSA and MSS2000. We will then discuss the processes and tools involved for development of IBFIS DBMS. Later we will explain how image data are stored and treated.

2. BALLISTIC FIREARM IDENTIFYING SYSTEMS

A BFIS refers the process of matching a cartridge to a weapon that is essential to solving many violent crimes. A spent cartridge case exhibits characteristic markings that can be used to identify the type, and possibly make, of weapon in which the cartridge was fired. A computerized BFIS simulates above operations and accumulates the data and images into the database.

Fireball [4] is one of the computerized BFISs. In the system the firearm ballistic images and information were stored in MSA to support Fireball applications.

	Residential Location		Street/ footpath		Open spaces/parks		Banks/ Credit union/ building societies		chemists/ pharmacies		Service stations		Other retail locations	
	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org	Ind	Org
Firearm	24.4	25.4	11.6	10.9	11.3	4.4	71.8	52.9	20	22.9	14.8	17.3	27.9	19.9
Knife/sharp instrument	17.3	34.2	42.8	49.4	16.9	54.3	9	16.1	68	35.4	60.6	43.6	19	40.3
Syringe	0.2	1.3	2.5	2.7	0	1.9	1.1	1.4	4	13.7	6.1	6.1	2	5.7
Blunt instrument	5.7	11.7	3.9	3.5	1.9	6.6	0.5	2.1	0	3.9	7.4	5.3	2.6	7
Other weapon	52.4	17.7	38.9	15.6	69.9	16.6	17.6	11.4	8	8.3	9.9	11.3	48.2	17.7
Imitation weapon/ threats	0	9.7	0.3	17.9	0	16.2	0	16.1	0	15.8	1.2	16.4	0.3	9.4

Fig .1*

As the level of violent crime involving firearms escalates, so too dose the volume of ballistics evidence that must be processed. Table 1 shows armed robbery by location, type of weapon used and type of victim in Australia [5]. Over time, the demands of FBDB application grow, become more complex, and need to support more users and Intranet applications. An Intranet BFIS system such as IBFIS is appreciated for these requirements. The database for IBFIS needs improvement in performance, scalability, security, reliability, recoverability, availability [6] and most importantly the Intranet accessibility, in comparison with its peer desktop applications. This is because of richer and securer data resources allow better chance of using existing information to identify the firearms. In addition to the fact that there are needs for a more refined

approach of obtaining images from the Internet [7], the IBFISDB enables an on-line ballistic/forensic image database.

The IBFISDB development also involves issues such as Intranet data processing, Intranet image processing, and Intranet programming and information security.

Fig. 1 shows IBFISDB table **tblHeadStamp**'s relationships with other tables.

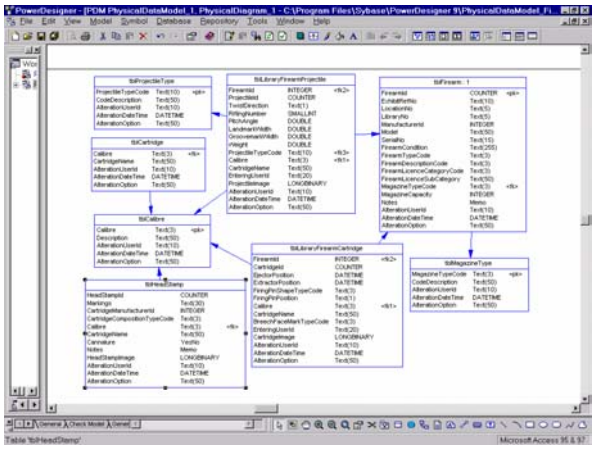


Fig. 1

3. COMPARISON BETWEEN MSA AND MSS2000

During the implementation of IBFISDB, it is found that MSS2000 improves application's performance, scalability, security, reliability, recoverability, availability and most importantly the Intranet integrity over the FBDB. We can make the comparison between these two DBMS products through FireBall and IBFIS. The major improvements include following respects:

Performance and scalability;
Availability;
Security;
Recoverability;
Reliable distributed data and transactions; and
Server-based processing.

High performance and scalability
IBFISDB offers better performance than FBDB. This is because MSS2000 provides support for very large databases, up to one terabyte, which is much larger than the current limit for an MSA database of two gigabytes. As large amount of images needed to be stored in IBFISDB, the limitation of size of the database becoming a critical condition for choosing database tools. Under the Windows NT system, MSS2000 works efficiently by processing queries in parallel and minimising additional memory requirements when more users are added. This high performance feature allows users to access images over the net in better speeds. In additional of above, stored procedures were used by IBFIS to speed up application's performance.

Increased availability
With MSS2000, IBFISDB can be back-upped dynamically, either incrementally or completely, while it's in use. Consequently, we do not have to force users to exit the database to back up data. This means IBFISDB can be running up to 24 hours a day, seven days a week.

Improved security
IBFISDB can integrate with the Windows NT operating system security to provide a single log on to the network and the database. This makes it much easier for administrator to admin complex security schemes. As IBFISDB is a server database, it is also better protected because unauthorised users can't get to the database directly without logging on the server.

Recoverability
In case of system failure (such as an operating system crash or power outage), IBFISDB has an automatic recovery mechanism that recovers the database to the last state of consistency in minutes, with no database administrator intervention. It is critical for an application like IBFIS can be up and running again right away.

Reliable distributed data and transactions
Transaction processing is a vital requirement for IBFISDB that is designed to support IBFIS critical applications, such as court evident and online matching and data entry. The visualized tool provided by MSS2000 for the tasks achieves this.

Consistency and recoverability of a database
In IBFISDB, each transaction is secured in the worse case of system failure and in the middle of complex updates by more than one user. This is because of MSS2000 treats all database changes inside a transaction as a single unit of work. By definition, either an entire transaction is completed safely and all resulting changes are reflected in the database, or the transaction is rolled back—and all changes to the database are undone.

Server-based processing
IBFISDB is designed as a client/server database. Data and indexes reside on the server that is often accessed over the network by many client computers of IBFIS. MSS2000 supports IBFISDB to reduce network traffic by processing database queries on the server before sending results to the client.

4. TRANSFERRING FBDB INTO IBFISDB

The development of IBFISDB is base on FBDB and it involves a set of activities as described in following.

Transformation of Schema
Evaluation of Modeling Tools: To transfer the FBDB structure and relationships between data, a FBDB Entity Relationship Diagram (ERD) needed to be generated by using one of database development tools. Amount them, Visible Analyst [8] (VA) and PowerDesigner [9] (PD) were examined.

VA was good at analyzing Primary Keys. It detects the tables that have not primary keys so that the keys can be specified or added in these tables. VA can also used to generate database schema in SQL document. This document can then be used later to create the tables in SQL Server 2000 (MSS2000) for Ballistic Firearm Identification Database (BFIDB).

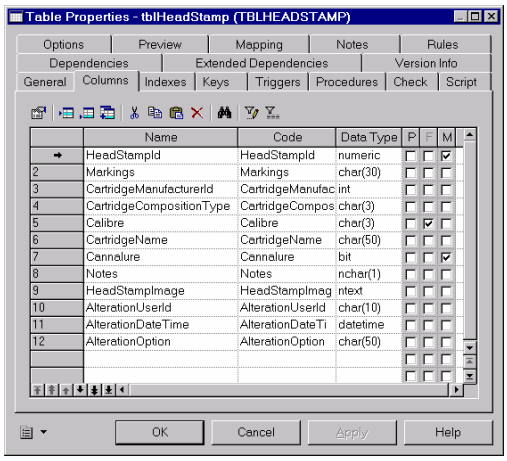


Fig. 2

PD shows clear entity diagram with additional information. The schema it generated specifies MSA's AutoNumber field as Counter type where VA has interrupted it into numeric data type with Identity property. PD seems more powerful and functional. Hence PD was selected for this instant.
Fig.2 shows the entity tblHeadstamp within the PD model for IBFISDB.

Reverse Engineering: By using the reverse-engineering techniques provided by PD, an ERD was generated based on FBDB. The tables, keys and references can than be analysis and specified. The necessary changes can then be made within the diagram.
After the examination and changes, a SQL scripts form of the schema is generated for MSS2000 to run to create the IBFISDB.

Primary Keys: To use a schema to create an IBFISDB database, to transfer the links existed in the original FBDB MSA database, and to maintain both data and referential integrities of the database, a primary key is required by each table. Two approaches were used to ensure each table has its key:
Specify key: For those tables whom have non-duplicate field(s), we can specify the unique field(s) as the primary key. MSA has a facility "find duplicates query wizard" which can be used to find if there is a unique component for a potential primary key.
Adds Key: For those tables that dose not seem to be organized around a unique component, we add a field named ID typed AutoNumber as the primary key for the table.

Generating Database*:* Once every table has a primary key, the database can then be reverse engineer again for creating the schema, and then generate the tables. It transfers FireBall OLE images into MSS2000 image data type. It is found that some data type was transferred wrongly from MSA to MSS2000. The data type *byte* was used by MSA and VA translated it as *Image* in MSS2000. Data type AutoNumber was translated into Numeric. This numeric data type will late be interrupted into MSA as Text(30), if such operation is required. PD dose better work in this area.

Create IBFISDB: After the analysis of the database, modification of the schema to suit the primary key, data type requirement, and references, a set of scripts is generated from the schema. By running of the scripts in MSS2000, the IBFISDB is created ready for facilitating the data. Fig. 3 shows part of scripts generated by PD to create the tables, keys and references for the IBFISDB.

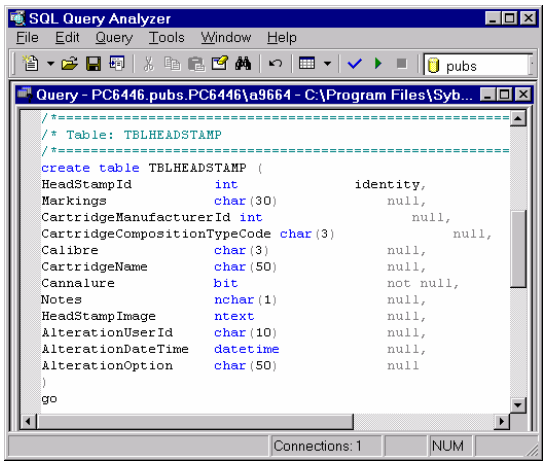


Fig. 3

Transformation of Data
Transferring data from FBDB to IBFISDB would be considerated as a relatively simple task. The MSS2000 Database Transformation Services (DTS) Wizard can be used for the purpose. However, it turns out to be a long and difficult process in the practice. First the transformation needs to follows the sequence specified by ERD. The parent data need to be in place before their children data. A DTS file is established to repeat this process. Next, while running this DTS file, an unexplainable random error message occurred during the transformation. It was found that it is caused by a bug that MSS2000 inherited from MSS97 together with Windows NT4. Because of a bug exiting in the product, however, this task turns out to be a difficult and long process. Different types of errors occurred randomly during repeated transformations. The problem is then referred to the network administrator. It is found that the errors were caused by a productions bug remained between Windows NT4 and MSS7.0 and inherited by MSS2000. By following Microsoft instruction of changing the security mode from Windows Authentic to Mix Authentic and use SQL standard login instead of Windows Trusted login, the errors were disappeared.

Examination of the Transformation
To ensure the data have been transferred completely, the PD is used again to reverse-engineer IBFISDB into a new ERD. The ERD is used to compare with the original FBDB to ensure the specifications are met. The IBFISDB is capable to be the backend for FireBall if there is such requirement. For this study, a completely new Intranet interfaced application is developed as IBFIS for Intranet users.

5. LOADING IBFISDB INTO FBDB

IBFISDB is capable to be used as the backend database for Fireball if it is desired. For those users who need to use the database in the network-isolated environments with their laptops, IBFISDB may be distributed into FBDB regularly using a set of Visual Basic compiled files. Because of the rich resource of the input and high standard maintenance, IBFISDB refreshes FBDB with most recent updated information to support the desktop BFIS Fireball. The desktop/laptop users may leave the database administration type of tasks to IBFISDB administrators and, on the other hand, to be benefited

from the rich resources of the Intranet database.
To distribute the IBFISDB into FBDB, the following activities would need to be scheduled into MSS2000 job manager:

Creation of FBDB
This task is performed by running a VB program which runs commands and scripts to create the FBDB tables, keys and references. The newly created FBDB will automatically alter some of the tables so that the database structure matches exactly to Fireball.

Distribution of Data
Same as the process transferring FBDB data into IBFISDB, the distribution needs to take steps following its ERD – parent data go first and child data go after. A set of VB exe files is used for this task.

Schedule of the Distribution
To catch up the current data in IBFISDB, above distribution needs to be scheduled to renew the FBDB regularly, said every 24 hours.

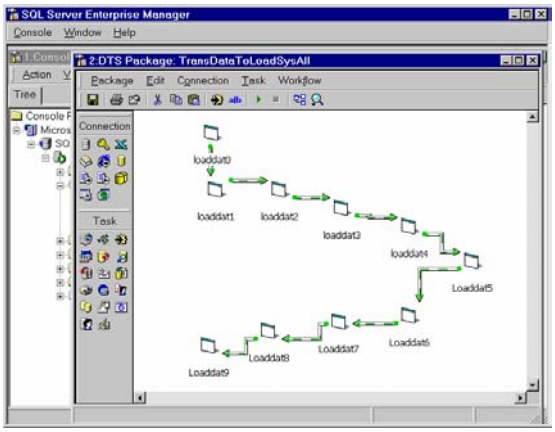


Fig. 4

Fig.4 shows the scheduled DTS for transferring data from IBFISDB to FBDB.

6. IMAGE DATA MANAGEMENT

Ballistic images are the major asset of the BFISs. They also create major challenge for this study's development. The transformation between IBFISDB and FBDB needs to take care the definition of the image data, security of the transformation and the accessibility of the data.

Image Data Storage
Image data in MSA are referred to as OLE (Object Linking and Embedding) data type. In MSS2000 they are referred as IMAGE data type and can also be called BLOB (Binary Large Object). In FBDB, the images save as Editor 3.0 Photo objects showed as Fig. 5. The image can be accessed by using MSA toolbox *bound Object Frame.*
When transferring FBDB into IBFISDB, the OLE objects can be transferred into Binary data automatically. However, these Binary data are not always accessible by its Intranet applications. To work around this problem, each of the images was first loaded into a temp file and late is uploaded into MSS2000 database. The loading and uploading programs are consistent therefore the images can be accessed consistently without concerning their types, size and etc.

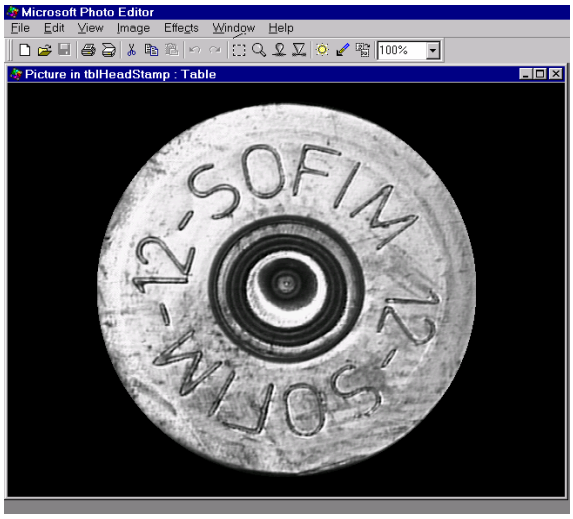


Fig. 5

With a new table option "text in row" provided by MSS2000, IBFISDB can also improve IBFIS's performance by stored the whole image into to the column without pointers when its size meets the conduction.

Image Assessment
In IBFIS, an ASP page uses following VB script to access object ADO [3] (ActiveX Data Objects) to load the image into the interface:

```
<%
Response.Expires = 0
Response.Buffer = TRUE
Response.Clear
Set cn = Server.CreateObject("ADODB.Connection")
cn.Open "driver={SQL Server};server=sciscjiao;" _
        & "uid=sa;pwd=;database=IntraBfis"
Set rs = cn.Execute("SELECT HeadstampImage FROM" _
        & "HeadstampImage WHERE HeadstampID=" & intId & ")"
Response.ContentType = "image/gif"
Response.BinaryWrite rs("HeadstampImage")
rs.Close
Set rs = Nothing
cn.Close
Set cn = Nothing
Response.End
%>
```

Fig. 6 shows a result of using ADO and VBScript to display a cartridge head stamp on an Active Server Page (ASP) using the Headstamp image stored in the IBFISDB.

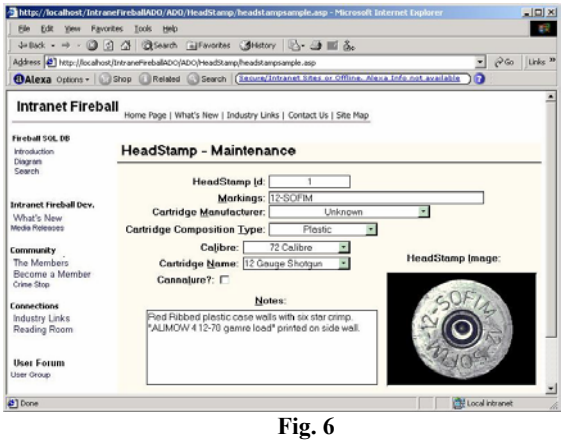


Fig. 6

7. CONCLUTION

Transferring FBDB to IBFISDB is part of the development

of IBFIS. This development not only provides an advance Intranet database for the users, but also allows the previous BFIS Fireball to be operated on a IBDISDB distributed MSA database in network isolated situations. This distributed database includes most recent ballistic information.

The limitation of the study is due to the time and tools restrictions as other DBMSs can be used instead of SQL Server 2000.

8. REFERENCES

[1] A. Australia Broadcasting Corporation Television, "Fireball," vol. 2001: Ian Watson, Naomi Lumsdaine, 2001.

[2] Microsoft, "Microsoft Access," 2000.

[3] Microsoft, "SQL Server 2000." Redmond, Washington, 2000, pp. 303-336.

[4] C. L. Smith, J. M. Cross, and G. J. Variyan, "Fireball: An Interactive Database for the Forensic Ballistics Identification of Firearms," *Research Report*, vol. 1995, 1995.

[5] J. Mouzos and C. Carcach, *Weapon Involvement in Armed Robbery*, vol. 38. Canberra: Australia Institute of Criminology, 2001.

[6] P. Atzeni and V. D. Antonellis, "Relational Database Theory." Redwood City Clifornia: The Benjamin/Cummings Publishing, 1995, pp. 250-255.

[7] R. Elmasri and S. B. Navathe, "Fundamentals of Database Systems," 3rd ed. Reading, Massachusetts: Addison-Wesley, 2000, pp. 324, 344.

[8] Visible, "Visible Analyzer," 2001.

[9] I. Sybase®, "Sybase Selected By Ncr To Improve Customer Service At The United States Postal Service," 1999.

Performance Comparison of Web-based Database Access

Gabriele KOTSIS
Department for Telecooperation, Johannes Kepler University
Linz, A-4040, Austria
E-mail: gk@tk.uni-linz.ac.at
and
Lukas TAFERNER
Abteilung für Wirtschaftsinformatik, Wirtschaftsuniversität Wien
Vienna, A-1090, Austria
E-mail: h9551164@gutemine.wu-wien.ac.at

ABSTRACT

In the recent decade the internet has gained importance as a medium for information and entertainment. Many 'old economy' enterprises discovered new possibilities to reduce e.g. advertisement costs taking advantage of this new medium. A lot of 'new economy' enterprises were founded in order to base their business model solely on advantages the Internet offers. In such an environment where turnover is generated by the users visiting a web site or purchasing goods or services using a web site, performance and uptime are crucial factors for being successful. As recent events have shown, a downtime due to hard / software failure or Denial of Service attacks causes the loss of transactions and turnover during the downtime and may inflict a considerable damage to a company's public image. In this paper we demonstrate a systematic approach for evaluating the performance of a web server. We will apply this approach in the analysis of an existing Web Site, serving mainly requests involving data base transactions. The objective of the case study is to identify the most performing technique for web-based data base access.

Keywords: n-tier-architecture, php, java, xsp, web benchmarking, performance of e-commerce services, case study.

1. INTRODUCTION

The World Wide Web (www), which originally was nothing more but a system allowing researchers to link static documents together by hyperlinks, turned out to be an interesting place for all types of companies to advertise their products on one hand and to interact with potential customers on the other hand. Hypertext Markup Language is due to its static nature not able to fulfil those needs. Powerful server-sided applications were needed in order to add interactivity to static web sites. The first well known evolved in the early 1990ies, the Common Gateway Interface (CGI), later on other powerful applications like PHP, ASP, JAVA and Cold Fusion where developed [1]. These applications provide the technology needed for E-Commerce, supporting interaction with database systems, presentation of content dynamically and responsiveness to users inputs. These applications are often very resource intensive, they have to retrieve information from databases, prepare and layout the information as desired and present it to the user. Especially web-servers of big companies such as Microsoft, Hewlett Packard, but also search engines such as Yahoo.com and Google.com have to serve a couple of million users per day, causing heavy load on the servers. The consequences for a website of delivering poor performance or being 'down', even only for a couple of minutes can be ruinous. The total loss caused by denial of service in 2000 amounted to almost 8.3 million US$ [Att00]. Considering these figures, which include only reported incidences, and adding the loss that cannot be measured (loss of customers confidence, loss of brand) make it worth to consider strategies to prevent those scenarios.
In this paper we will have a closer look at the factors influencing web server performance and the options a Webmaster has to find the most suitable solution for a web project. First, we will explain the principles and concepts of web server performance evaluation, and introduce the components that are responsible for the performance of a web application. In the next section, the case study used in the evaluation will be briefly presented. The major part of the paper is dedicated to the description and interpretation of the performance tests. We conclude with a summary of results and some general insights that could be derived from the case study.

2. BASICS IN WEB PERFORMANCE EVALUATION

A typical (web) performance evaluation life cycle [2] starts with the definition of the system under study and the goals of the study along with a specification of the performance metrics to be considered. In this work, we will focus on multi-tier architectures, which are typical for nowadays web-based e-commerce sites. This architecture can be represented as shown in Figure 1. The main components to be considered in the performance evaluation are the users´ client, the network connecting clients and servers (typically the internet), the web server, the web application server, the DB management system as well as the network linking the web server, the web application server and the DB system if those system components do not reside on the same machine.
The first performance metrics of interest is response time, which is defined as the time between the submission of a request from a user and the completion of the response (i.e. the instance in time when the client has received the response). This measure is a critical factor in the evaluation of web services. It is known from empirical studies, that if response time exceeds 8 seconds, users tend to terminate the session and will leave the site (eight-second rule, [3]). Such a bound set upon a performance metric is called "service level agreement". The second important criterion is throughput, which is defined as the number of requests the system can complete per unit time. A typical question in web service capacity planning is to identify the maximum throughput that the system can achieve under a given service level.
Both measures are related to the "speed" at which requests can be completed, either seen from the users´ point of view (response time) or from the system point of view (throughput). Other criteria include scalability, which is defined as the

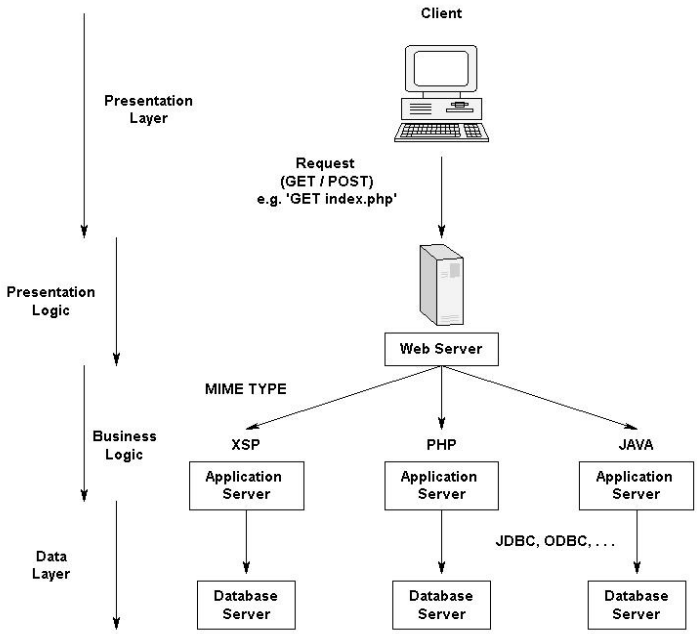


Figure 1

systems ability to handle an increase in the load, and availability, which is defined as the ratio of time the system was up serving requests to the total time (up and down times).

3. CASE STUDY

The application that is taken to test the speed, scalability and reliability of different components of dynamic websites is the homepage of the Vienna City Marathon. This marathon is carried out in three different heats starting at the same time: the marathon (the traditional 'Marathon' - distance has to be mastered), the spring run (15,4 km) and the marathon relay where the marathon distance is done in teams. The purpose of the website is to offer real-time results online via HTTP and WAP. While those services are offered free of charge to the customers, the architecture of the site is also typical for commercial services and therefore serves as a representative test-bed. The reason for choosing this site is, that the system can be observed under real, heavy load, namely at the day of the race, where several thousand database requests are submitted from all over the world. In this paper only the HTTP access is considered because it is the most important part of the application in terms of performance, the number of requests via the wireless application protocol (WAP) is not significant.
In the following, we will discuss the different components of this test site briefly.

Database

The database for the marathon project consists of three tables corresponding to the three heats mentioned before. These tables contain similar information, namely the result records for the respective participants. One significant difference is the size of the tables: marathon contains about 10000 records (runners), whereas relay and spring amount to about 4500 entries. The different table sizes will allow us to study the effect of the table size on system performance.
The database used in the project was a MySQL database server [4]. As only simple queries (selects involving no joins and only rarely update statements) are expected, this database delivers sufficient performance, there was no need to use a more powerful commercial product such as an oracle database (performance tests were performed to verify this conjecture, the performance was mainly dominated by the amount of memory available at the physical machine, not by the type of DBMS used).
The DBMS is installed on a standalone machine running SUSE Linux, it is important to provide enough RAM to keep the databases in the cached memory to increase speed. Furthermore, MySQL has to be compiled for heavy duty and with specification of the cache memory. To provide equal conditions to all web server applications, this MySQL configuration will be used in all three tests. There may be differences in connection speed however, depending on the quality of ODBC and JDBC drivers available for the platforms tested.

Web server Application

The objective of this paper is to test the performance of different web server applications, much attention has to be given to developing the server sided scripts and servlets. Three very different technologies were chosen to provide an overview and to detect weaknesses and advantages of different solutions.

JAVA servlets [5] are a fairly new technology, they provide a way to enhance the functionality of a web server. The advantage of servlet technology results from the fact that the (JAVA) source code is already compiled at runtime. This implies the advantage of 'Write Once, Run Anywhere', i.e. the byte code can be transferred across operating systems and web servers. To run servlets in a web server environment a servlet container is needed, which acts like a HTTP server and invokes the servlet when requested. In this work, the Open Source Tomcat Servlet Engine was used, which is considered to be the most mature platform. The servlets were developed within J2EE (Java 2 Enterprise Edition).

Cocoon/XSP [6] is a pure Java publishing framework that relies on new W3C technologies (such as DOM, XML, and XSL) to provide dynamic web content. It offers a different way of working, allowing content, logic and style to be separated out into different XML files, and uses XSL transformation capabilities to merge them. Cocoon provides an environment that allows a server side embedded script language to be executed known as XSP (Extensible Server Pages). The server is able to contact a DBMS via JDBC and can retrieve data which are stored in an XML document. This document can be cached by the server and is sent to the clients Web Browser as plain HTML.
The actual implementation for this study was built on a cocoon 2.0 platform using the same Tomcat installation as mentioned above in the section on Java/Servlets.
PHP [7] is probably the most popular server sided scripting language. Basically one has two options for how to install PHP on a Apache web server: as a standalone executable that is executed on demand or as a compiled module of the web server that is loaded at start-up. For the purpose of benchmarking PHP it is better to compile mod_php (4.0.6) into the Apache web server (1.3.22) because it is said to be more stable and reliable [7]. PHP has a built in MySQL support, no additional driver or interface is needed. In contrast to JAVA and XML applications, PHP is not a compiled language but is compiled at run-time. We shall see in the performance tests to which extent this behavior affects

performance.

Network Finally we have to consider the importance of bandwidth as a factor in evaluating web server applications. Bandwidth is usually measured in bps (Bits per second) that can be transported at a given time over a TCP/IP connection. The average size of a request can be estimated to be 170 Bytes. The average size of a response amounts to approximately 4450 Bytes (both figures were derived from measurement experiments). In total, one transaction requires therefore approximately 4620 Bytes to be transferred. This number has to be compared to the bandwidth available during the test. All benchmarks were carried out on a 100mbps network. Due to overlay effects that occur in any network, the technical possible throughput is decreased by 20 per cent, i.e. 80mbps (equivalent to 10 MB per second) could effectively be used. The result of this simple calculation shows that more than 2000 transactions can be handled per second. If this number is exceeded, than the network will become a bottleneck. Note that the network can in fact become a significant performance bottleneck considering the capacity offered by the Internet.

4. PERFORMANCE TESTS

User behavior

To standardize the test and invoke the same balance on all three implementations of the Vienna Marathon Website it was necessary to have a close look at the log files of previous competitions, where server load was considerable. After an analysis the average user behaviour could be reduced to six different types of information a user would request:

(1) Search for the first name of a runner in the standard interface (e.g. All runners whose name is 'Michael')

(2) When the person desired is found in the resulting list, a search for the start number of this person is requested (e.g. Startnumber = 1201)

(3) In the third step the visitor will have a closer look at the results of the runner selected and proceed to the page with detailed results, where average speed and results are displayed.

(4) Next request will be submitted using the detailed search form looking for all runners that are from a specific country (e.g. Nation = AUT)

(5) Another point of concentrated interest is to have a look at the first couple of positions in a heat (e.g. Position < 100)

(6) Finally, the visitor may be interested in the fastest runners and may request a list of all runners, that completed the first half-marathon faster than a specific time (e.g. 1.HM < 1:30)

Between the requests an average user think-time (i.e. the time between displaying the requested results and the issuing of a new request,) was set to two seconds.

Sequential Tests

In the first test environment the objective was to test the web server applications in a sequential test, i.e. requesting the same query 100 times after each other, to see how the three applications compare to each other. In addition, the size of the database was varied to get insight about the amount of time that is consumed by requesting data and processing data.
The web server (Apache 1.3.22, Tomcat 4.0.1, Cocoon 2.0.1) and the DBMS (Mysql 3.23.47) were installed on the same system (Intel Pentium II, 399 Mhz Processor, 256 MB RAM) running SUSE Linux 7.3. The requests were submitted from another machine acting as the client.
To avoid caching effects, the search string in the first user request (search for the name of a runner) is fetched at random from a pool of different strings to represent a more realistic load.

Response Time The results of the first sequential tests were not quite surprising: Java was supposed to be the fastest in terms of response time, PHP second and XSP last. Detailed results can be seen in Figure 2. The graph shows the sum of response times of all tests conducted in comparison to each other. As expected, the Java application behaved best under stress, performance did not significantly decrease with increasing size of the Database. PHP, which had a good start, however, increased its total response time faster than database growth, which is an indicator of poor scalability.

Drill Down Tests

To gather more information about the behavior with even more load, several drill-down tests were conducted to identify the system capacity, i.e. to find out how much load the three applications can handle and how they perform under an

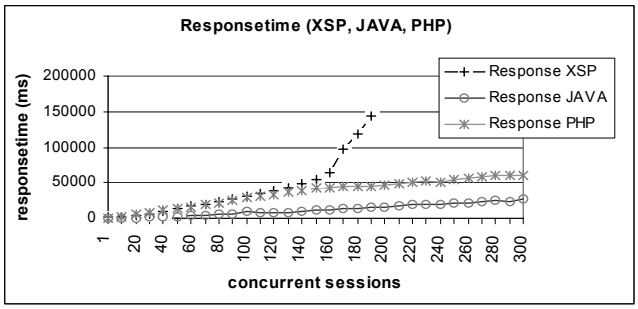


Figure 2

increasing amount of requests. The benchmarks were carried out on the same computers (i.e. identical hard- and software) as described in the chapter sequential tests.

Scalability In a first series of tests, httperf [8] was used to benchmark the system and to evaluate and compare the response times of the three different application server techniques.

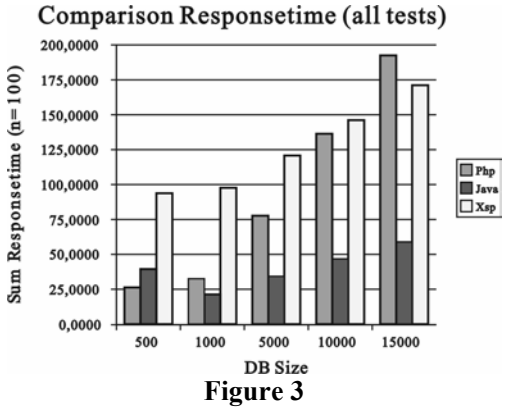


Figure 3

3 shows the average response time measured during each httperf drill down benchmark, comparing all three technologies to each other. As predicted, JAVA servlets show by far the best performance relative to the other technologies,

PHP comes second place with almost double response time (granularity: the number of concurrent sessions was increased by 10. The benchmarks were carried out without breaks between the single tests in order not to provide relief to the server, i.e. starting a standalone test with e.g. 200 concurrent sessions would lead to another result as shown here). Worst performance could be monitored during the XSP tests, at a level of about 150 concurrent sessions cocoon crashed and left the server in an uncontrollable state. After the test was repeated several times with unchanged result, I decided to leave out XSP for further tests due to its uncompetitive nature concerning performance under heavy load.

Availability In a second series of tests, http_load [9] was used, which runs multiple http fetches in parallel, to test the

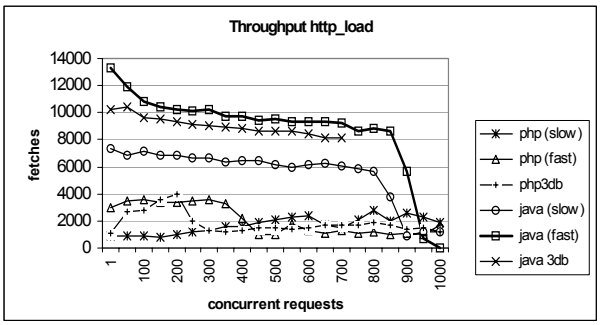


Figure 4

throughput of a web server. However unlike most such test clients, it runs in a single process, so it doesn't bog down the client machine. The results obtained were similar to the results obtained with httperf. In addition, several benchmark sequences were carried out with different hardware sets. The machine which is running tomcat and the apache – php module was kept the same, the main aim was to find out, to what extent the database performance would influence the total response time. Three different hardware setups were tested:

- The web server has to fetch database result tables from one mysql machine with weak hardware (Pentium II 200 Mhz, 256 Mb RAM).
- The web server has to fetch database result tables from one mysql machine with strong hardware (Pentium IV 2000 Mhz, 512 Mb RAM).
- The web server has to fetch database result tables from three different machines (Pentium II 200 Mhz, 256 Mb RAM; Pentium III 500 Mhz, 256 Mb RAM; Pentium IV 2000 Mhz, 512 Mb RAM) and manage connectivity to all three mysql machines.

The benchmark using http_load was carried out several times on every hardware setup. This test was used to test the availability in counting the number of observed errors when serving requests. Again, Java showed best performance (see , where (slow) means the results of the benchmarks using the slow mysql machine, (fast) indicates the usage of the strong hardware for the database machine and 3db represents the results of three hardware databases).

The Java servlet implementation does not produce any errors (and seems to be independent of database speed) at all up to approximately 700 requests per second, reaching a level of requests per second where the application becomes unavailable (a restart of the server is necessary). PHP however, starts producing errors from the very beginning of the test, the error rate increases with requests per second and seems to be depending on the speed at which database result tables are served.

Throughput The most interesting result is the total throughput that can be produced by all three applications compared to each other. These results are shown in. It is obvious that java servlets gain most advantage using one fast database machine, the handling of three machines containing one slow compute decrease performance considerably. Php produces a lot of errors as soon as performance is critical due to high requests per second. To sum it up, php's performs by far not as good as java does, this effect is strengthened if the web server undergoes heavy load.

Real Life Test After identifying the Java Servlets implementation as the best performing method, we decided to use this technology for the data base queries for the Vienna City Marathon 2002. An array of server was used to cope with the expected load on the day of the race and on the days after. The architecture used is shown in Figure 7.

The main load was supposed to be taken by the two machines gretchen and zerberus which were the best machines in terms of hardware. The machine called snoopy was the mysql master for four mysql slaves (i.e. UPDATE, INSERT and DELETE on the master causes the same action on all slaves) and carried out the updates coming from time measurement and organized the import to all other databases.

Figure 5 summarizes the total number of database requests served under real load. Response time was within the range of seconds for the average requests, but sometimes the service level of 8 seconds could not be met despite of the pool of servers in cases of heavy load (requests arriving in large bulks when popular runners are finishing). Nevertheless, from a performance point of view the Vienna City Marathon was completed successfully.

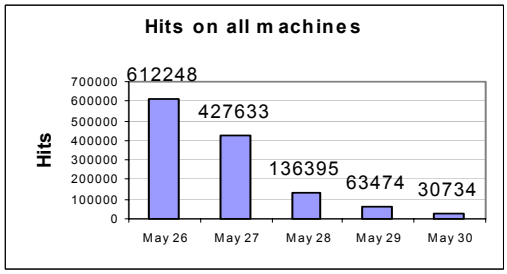


Figure 5

5. CONCLUSIONS

As we have seen in the benchmarks it is vital to make a good choice which web server application to use. This depends heavily on the circumstances and desired properties of the web server application. The chart in Figure 6 provides an overview about which web server application can be used in terms of load it can manage.

	Load Level		
	Light < 1 req sec	Medium < 200 req/sec	Heavy > 200 req/sec
XSP (cocoon) SLA	X < 2 sec	-	-
PHP (apache) SLA	X < 1 sec	X < 15 sec	-
JAVA (tomcat) SLA	X < 1 sec	X < 8 sec	X > 8 sec

Figure 6

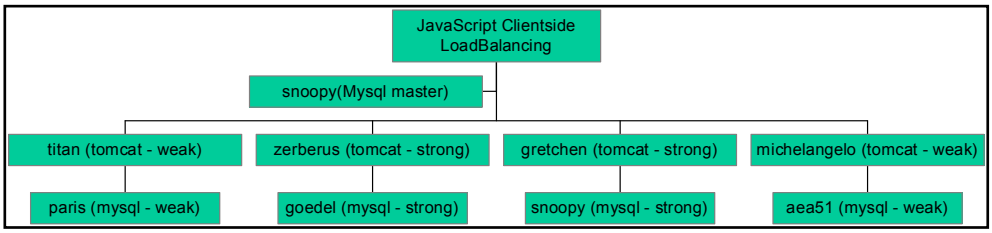


Figure 7

From the results of the benchmark series the following conclusions could be drawn:
XSP is in an early state of development to be competitive. Furthermore documentation was virtually non existent and the implementation process (separation of content, style and management) is too complex for small projects.
PHP showed acceptable performance during the benchmarks, could, however, not beat JAVA. Its advantages are a comprehensive language and easy deployment and configuration of the server platform. A PHP-based solution can be recommended for small-sized enterprises expecting low to moderate load, where the major concern is low development and maintenance costs.
JAVA servlets performed best compared to the other technologies, in the sequential tests as well as during the drill down benchmarks. It is by far the most professional tool available but on the other side difficult to learn and to deploy.
The results of this case study can be generalized to other application domains as well. The assumption in this case study was a specific pattern of the load submitted to the server. The database was mainly serving select statements of different complexity. There was no significant load from updates or delete statements. This is a typical load pattern for information systems, where the purpose of the underlying database is mainly to store information which is basically entered once for frequent subsequent retrievals. It is in general not suitable for e-commerce applications like electronic shopping malls, where users would typically purchase goods and the purchase data must be entered into the database. But it might be a representative model for the information retrieval phase of the business transaction.

6. REFERENCES

[1] Krause, Jörg: PHP, Webserver Programmierung unter Windows und Linux, 2000, Carl Hanser Verlag, Munich.
[2] Jain, Raj: The Art of Computer Systems Performance Analysis, 1991, John Wiley and Sons Inc.
[3] Menasce, D., Almeide, V.: Capacity Planning for Web Services, Prentice Hall PTR, 2002
[4] MySQL Database Server, Product Description, http://www.mysql.com/
[5] Callaway, Dustin R.: Inside Servlets, Second Edition, 2001, Addison Wesley Verlag, Munich
[6] Cocoon Developers Homepage, http://xml.apache.org/cocoon/
[7] PHP Documentation and Download, http://www.php.net/download
[8] HTTPERF Benchmarking Tool, http://www.hpl.hp.com/personal/David_Mosberger/httperf.html
[9] HTTP_LOAD Benchmarking Tool, http://www.allan.org/load/http_load/

Binary Shortest Path Routing for Congestion-Driven Max-Min Fairness

Cao Jing
Department of Computer and Information Engineering, Hohai University
Nanjing,210024,China
E-mail: wellindustry@sina.com

ABSTRACT

The control scheme on SPR(Shortest Path Routing), Max-Min Fairness and RM feedback cannot prevent an ATM network from congestion. Integrating Divide-and-Conquer and SPR together, a BSPR(Binary Shortest Path Routing) is obtained in this article, which will find the least congestion and the shortest path. The algorithm's total complexity is $O(\log_2^n)$ times the complexity of SPR. So, it can be applied to ATM or other rate-based public networks.

Keywords: BSPR, Max-Min Fairness, Congestion, Flooding.

1. INTRODUCTION

SPR is due to Dijkstra(1959), which is applied to Internet as one kind of TCP/IP routing protocols—OSPF(Open Shortest Path First).

ATM forum has chosen the rate-based flow control framework for ABR services [1][2].

The well-known Max-Min Fairness definition was introduced and discussed in [3,4] and has been adopted as the desired ABR allocation criterion by the ATM Forum [5]. However, computing exact Max-Min rate on a general topology network under the assumption of no intermediate buffering and minimal communication delay has been shown to be NP-hard [6].

Developing an approximate algorithm of max-min fairness has received a considerable amount attention recently ([7-9]).

The RM cell[1][2] (carrying the traffic information of the selected path) is sent back along the path form destination to source, this feedback signal is taken to control the source to increase or decrease the sending rate. The control scheme may prevent congestion and get a maximal output within this connection though sending RM cell back periodically also costs much communication resource.

However, even max-min fairness and RM feedback control scheme can't prevent congestion though the path selected by SPR has least communication delay and is shared among users fairly, it may also be the busiest one over the whole network.

Many congestion-driven fairness routing algorithm [10][11] have been obtained now.

The conflict graph, spanning tree, maximal clique, direction and other definitions were introduced to study congestion-driven fairness algorithm in [10].

A TL(Transfer Line) model is introduced in article [11] to study routing, queue length, utilization and other Q.o.S issue of ATM or other rate-based network. Some results had been obtained.

Definition of TT Rate

Transferring and transmitting are two main independent operations in any ATM switch, they send cells from request queue to output queue and from output queue to output link consecutively. Their rate depend on how many CPU time slots and how much output link bandwidth a TDM (Time Division Multiplex) switch allocates to the request.

Transferring and transmitting operation are simply called as TT operation and their rate as TT rate.

Queue Control

Suppose that $\theta = \max_{1\le\varpi\le k}\{\nu_\varpi\}$, where ν_ϖ is the TT rate the request has obtained at ϖ th switch within this connection.

(1) If the source transmits cells at a rate ν that is greater than θ, there must exist overload and congestion at a node within this connection.

(2) If $\nu \le \theta$ and the buffer size at each switch is managed no lesser than L_{n_0} [11], there will not exist any congestion within this connection.

(3) If $\nu < \theta$, the connection will also have resistibility from jitter, the margin is $\nu_\varpi - \nu$ at switch ϖ and the queue length will become shorter and shorter.

So, we call θ Connection Bottleneck pertaining to this request and the node ϖ such that $\nu_\varpi = \theta$ Bottleneck Node.

Delay Computation

When the control scheme is rate-based, $\nu \le \theta$, the absolute delay from source to destination is not the propagation delay $\sum p_i$, but as:

$$\frac{2\times k}{\nu} + \sum_{i=1}^{k} p_i \qquad (1)$$

Where k is the hop number, p_i is the propagation time over i th transmission media.

The relative delay is:

$$\frac{1}{\nu} \qquad (2)$$

$1/\nu$ may be little when the switch is idle and can be omitted in Exp. (1), but it will become larger and turns into a key factor in expression (1) and (2) especially when the network tends to have a congestion or is busy.

A Congestion-Driven Adaptive Routing Algorithm

A congestion-driven adaptive routing algorithm is obtained [11]. The simulation illustrates that the algorithm does not insist on going along SPR, but chooses routes adaptively according to the network's congestion level. It tries to direct the cells going along nodes and paths those are less busy. It balances the

resource utilization and reduces congestion over the whole network.

However, the algorithm is not implemented in [11].

Integrating Divide-and-Conquer and SPR together, an advance BSPR(Binary Shortest Path Routing) algorithm is obtained in this article, which will find the least congestion and shortest path.

The algorithm's total complexity is $O(\log_2^n)$ times the complexity of SPR. So, it can be applied to ATM or other rate-based public networks.

2. THE PROBLEM OF MAX_MIN ROUTING

The issue of fairness is intimately related to the way network service providers manage their networks in term of service billing and other complex network optimization problem.

Definition 1 The policy according to which a switch ϖ allocates its communication resource among requests is called Resource Allocation Policy of ϖ, simply as Policy.

Definition 2 The communication resource a switch ϖ allocates for the request *R* is called AR(Available Resource) *R* having obtained at ϖ, or the affordable resource ϖ providing to *R*. Simply, denote it $R(\varpi)$.

Definition 3: If there is a path $P=\varpi_1\varpi_2,\varpi_2\varpi_3,...,\varpi_{k-1}\varpi_k$ from ϖ_1 to ϖ_k in the network G, where $\varpi_1,\varpi_2,\varpi_3,...,\varpi_{k-1},\varpi_k$ are the vertices on P. Let:

$$R(P)=\min_{\varpi\in P}\{R(\varpi)\}$$

Then $R(P)$ is called Available Resource *R* having obtained on *P*.

Definition 4 Suppose that *S*, *D* are the source and destination of the request, $R=\{P\,|P$ is a path from *S* to *D* in $G\}$.

Let:

$$R(G)=\max_{P\in R}\{R(P)\}=\max_{P\in R}\{\min_{\varpi\in P}\{R(\varpi)\}\} \qquad (3)$$

Subject to:

$$PDelay(P)+\frac{2\times Hop(P)}{A(P)}\le\pi \qquad (4)$$

Where $PDelay(P)$ is the propagation delay of P, π is the delay bound demanded by Q.o.S. Then $R(G)$ is called Available Resource *R* having obtained in *G*.

Definition 5 The path *P* such that $R(P)=R(G)$ is called MMP (Max-Min Path) and the MMP that has the least delay is called SMMP (Shortest Max-Min Path). The algorithm to find SMMP is called MMR(Max-Min Routing).

The "max and min" operations in Exp. (3) illustrate that MMR selects path for *R* with another "Max-Min Fairness" to prevent congestion.

Lemma 1: Let P, $\overline{P}$ are two paths from *S* to *D*, $P\supseteq\overline{P}$ (*P* contains more nodes than $\overline{P}$), then:

$$A(P)\le A(\overline{P})$$

So SMMP will have no loop and its length is lesser than d, where *d* is the diameter [10] of *G*.

Lemma 2: If the available resources of all paths from *S* to *D* are constant, the delay expression will be linear over those paths.

Example 1 : Given a public ATM network whose topology graph is as Fig.1. CS(Cell/Second, 1CS≈13bps) is taken as the metrical unit of resource. The total resource at each node in {C, E, F, G} is 30000 CS, 60000 CS in {B, H} and 90000 CS in {A, D}.

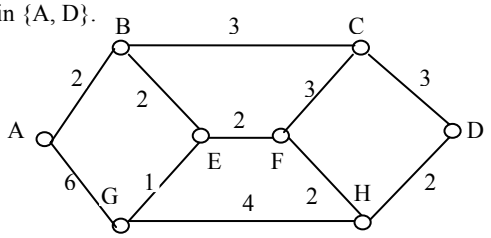


Fig.1 The network's topology

All nodes have the same policy: all requests have the same priority and can take up δ of the left amount.

$$\delta=\begin{cases}0.4 & \rho\le 0.4\\ 0.3 & 0.4<\rho\le 0.6\\ 0.2 & 0.6<\rho<1\end{cases}\quad \text{if}$$

Where ρ denotes the busy rate at that node.

Along comes a request *R* from *A* to *D* and its Q.o.S demand on delay is lesser than 17 ms(milliseconds) .

Table 1 Available resource the request R applied at each node in *G*

Node	A	B	C	D	E	F	G	H
ρ	.8	.8	.9	.8	.8	.8	.8	.8
AR	3600	2400	600	3600	1200	1200	1200	2400

Table 1 illustrates the available resources *R* having obtained at each vertex in *G*. So:

1) The paths from *A* to *D* in the network are P_1= 'ABCD', P_2= 'ABEFHD', P_3= 'AGHD',…, and so on.
2) The available resources *R* on P_1,P_2,P_3 are 600,1200,1200 respectively.
3) The available resource of P_2 and P_3 are the same as $R(G)$, but P_2's delay is bigger than P_3 though its propagation delay is lesser than P_3, so P_3 is the SMMP for this request.
4) If $\pi=15$ ms, P_1 will be the SMM

3. BSPR ALGORITHM

In this section, a static BSPR routing algorithm is given to find SMMP.

BSPR Algorithm

Subroutine Extended_SP(C, S, D, G, π)

{

1) Delete all nodes whose AR is lesser than C and all adjacent edges of them in G, denotes the left graph G';

2) If there exists a path P from S to D that is the shortest one in G' but the cost function is $PDelay(P)+\frac{2\times Hop(P)}{R(P)}$, return true; otherwise, return false;

}

BSPR(S, D, G, π)

{

1) Compute $R(\varpi), \forall \varpi \in G$;

2) Put them in an array Ω, keeping its elements distinct each other and in increase order;

3) Low=1,High=$|\Omega|$;

4) while Low ≤High do {

C=Ω[$\lfloor$ (Low + High)/2 $\rfloor$];

if Extended_SP(C, S, D, G, π)

Low=$\lfloor$ (Low + High)/2 $\rfloor$+1;

else

High=$\lfloor$ (Low + High)/2 $\rfloor$-1;

5) The ***last*** P that makes Extended_SP(C, S, D, G, π) true will be the ***SMMP***.

}

Proof: If P is the SMMR of G, there will exist an integer i such that $R(P)=R(G)=\Omega[i]$. The delay is linear by **Lemma 2,** so P is the SMMR, with the same proof of dijkstra SPR.

4. PERFORMANCE EVALUATION AND SIMULARION RESULTS

Example 2: Route with the BSPR in the ATM network, which is the same as Example 1.

Ω contains two elements, 1200 and 600. Fig.2 is the left graph after deleting the nodes whose AR is lesser than 1200. Fig.3 and Fig.4 demonstrate the process of BSPR. P_3 is the SMMP to this request.

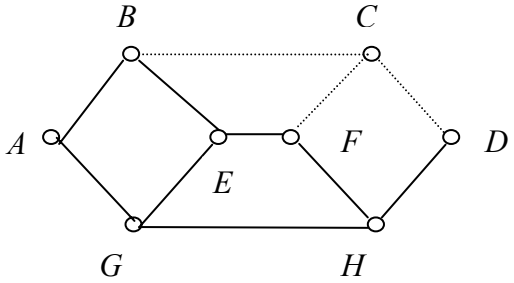


Fig.2 The left graph after executing step 1 of Extended_SP

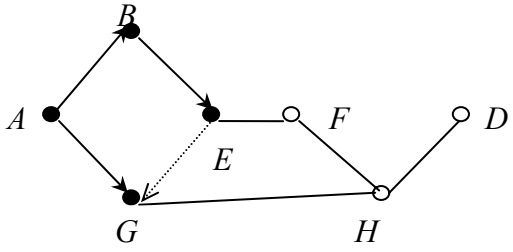


Fig.3 One graph of executing step 2 of Extended_SP

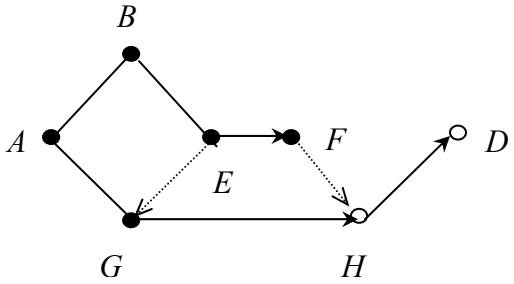


Fig.4 The graph after finishing step 2 of Extended_SP

Table 2 The process of MMR

Times	SMMP	$R(P)$	ρ_A	ρ_B	ρ_C	ρ_D	ρ_E	ρ_F	ρ_G	ρ_H
1	P_1	7200	.48	.52	.64	.48	.4	.4	.4	.4
2	P_2	7200	.56	.64	.64	.56	.64	.64	.4	.52
3	P_3	7200	.64	.64	.64	.64	.64	.64	.64	.64
4	P_1	2160	.664	.676	.712	.664	.64	.64	.64	.64
5	P_2	2160	.688	.712	.712	.688	.712	.712	.64	.676
6	P_3	2160	.712	.712	.712	.712	.712	.712	.712	.712
7	P_1	1728	.7312	.74	.77	.7312	.712	.712	.712	.712
8	P_3	1728	.75	.74	.77	.75	.712	.712	.77	.74

Example 3 : Given a public ATM network whose topology is the same as Fig.1. The initial busy rates of all nodes in the network are 0.4, other important factors are the same as **Example 4 :**Along come 10 requests at the same instantaneous from A to D and all requests have a same Q.o.S demand in delay, $\pi \le 25$ ms (milliseconds).

Table 2 illustrates:

1) The algorithm chooses route truly and adaptively basing on the network's traffic. It does his best to make the data go along the idle paths and nodes. So in this example,P_1 P_2, and P_3 share the load. The busy rates of all nodes are equal when the network having served the third and the sixth requests and the system achieves an absolute balance.

2) When serving the eighth request, MMR makes the data going along P_3, not P_2. The delay of P_2 is beyond the Q.o.S demand.

3) Routing with SPR and allocating resource with max-min fairness[2],all nodes in P_1 will be busy while the other nodes of G will be idle. Anyway, the total output with SPR is 1/3 of SMMR and the refusal probability is much bigger than SMMR with the same max-min fairness policy.

Theorem 1: In the worst case, the complexity of BSPR is $O(\log_2^n)$ times the complexity of SPR.

Proof: The dijkstra shortest path algorithm will be executed $\log_2^n$ times at most.

Theorem 2: The time delay arose by BSPR will be lesser than $\log_2^n \times \min(\pi, \delta)$, which is independent of propagation delay.

Theorem 3: When $R(G)$ is equal to $\max\{R(\varpi) \mid \varpi \in G\}$,

the SMMP will be the same as the shortest path when $PDelay(P)+\frac{2\times Hop(P)}{R(P)}$ acts as the cost function.

4. CONCLUSION

BSPR routing algorithm is obtained in this article. It integrates resource allocation, routing and flow control into a close-loop architecture adaptively. It prevents congestion, reduces the refusal probability and increases the output eminently. It also can control delay and delay jitter. The algorithm relies on SPR and its complexity is $O(\log_2^n)$ times the complexity of SPR, so it can be applied to ATM or other rate-based networks.

5. REFERENCES

[1] A.Charny et al., Time-scale analysis and scalability issues for explicit rate allocation in ATM network, IEEE/ACM Trans. Networking, Vol. 4(8), 1996, pp569-581.

[2] Flavio Bonomi and Kerry W.Fendick, The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service, IEEE Network, Mar./Apr., pp25-39,1995.

[3] D. Bertsekas and R. Gallager, Data Networks, Second Edition. Englewood Cliffs, NJ: Prentice-Hall, 1992.

[4] J. M. Jaffe, Bottleneck flow control, IEEE Trans. on Commun., VOL. COM-29, July 1981, pp 954~962.

[5] ATM Forum, Traffic management specification version 4.0, in ATM Forum-TM 95-0013R8,Aug. 1995.

[6] A. Bar-Noy, A. Mayer, B.Schieber, and M. Sudan, Guaranteeing fair service to persistent dependent tasks, presented at the Proc. ACM-SIAM Symp. Discrete Algorithms,1995.

[7] N.Ghani and J.W.Mark, Dynamic rate-based control algorithm for ABR service in ATM networks, in Proc. IEEE Globecom 1996, Vol.2, London, UK, Nov. 1996, pp 1074~1079.

[8] Nasir Ghani and Jon W.Mark, Enhanced Distributed Explicit Rate Allocation for ABR Services in ATM Networks, IEEE/ACM Trans. on Networking, VOL. 8(3), 2000, pp71~86.

[9] S. Kalyanaraman, Sonia Fahmy and Rohit Goyal et al., The ERICA Switch Algorithm for ABR Traffic Management in ATM Networks, IEEE/ACM Trans. on Networking, VOL. 8(1), 2000, pp87-98.

[10] Alain Mayer, Lyoram Ofek and Moti Yung, Local and Congestion-Driven Fairness Algorithm in Arbitrary Topology Networks, IEEE/ACM Trans. on Networking, VOL. 8(3), 2000, pp362-371.

[11] Cao Jing and Chen Shuzhong, The Model of Transfer Line and an Adaptive Congestion-Oriented Routing Algorithm, Science in China (Series F), Vol.44, No.4, Aug. 2001, pp270-277.

[12] A.S.Tanenbaum, Computer Networks. 3rd Ed. Prentice-Hall, 1996

Application of Web and Data Warehouse Techniques in DSS

Yongzheng Lin Kai Wang Bing Shi
School of Computer Science & Technology, Shandong University
Jinan, ShanDong 250061,China
E-mail: yzdynasty@sina.com

ABSTRACT

Web and data warehouse (DW) techniques, which appeared in recent years, make decision support system (DSS) further develop. Considering the problems of the original DSS, the paper summarizes the techniques of DW, presents the architecture of the DSS based on Web and DW. Finally, the realization methods and key techniques are discussed.

Keywords: Decision Support System (DSS), Data Warehouse (DW), Web Technique

1. INTRODUCTION

Decision Support System (DSS), which expands management information system, is a human computer interaction system which synthetically applies multidisciplinary knowledge from some other fields including computer technology, management science and artificial intelligence and so on. By means of current information techniques, it helps the managers make correct decisions to some semi-structure decision problems by offering scientific schemes.

Traditional DSS shown in figure 1[1] is normally composed of database, method base and model base. To date, it can't afford powerful function for compound, analysis and synthesis of data. However, DSS demands integrated, historical and synthetical data, which not only reflect instant state of changes but also show the history and the trend of things. It has been out of the range of what database system can do. The data warehouse (DW) techniques promote the new development of DSS. On the one hand, DW techniques separate information required for DSS from original operational data, on the other hand, they convert the scatted and difficult to access data into uniform information, which is available at any moment. Moreover, the developing web techniques afford a convenient way of gaining a lot of timely information and take on a friendly user interface. Hence, this paper puts forward decision support system based on web and data warehouse techniques so as to develop and perfect advance the development of DSS theory.

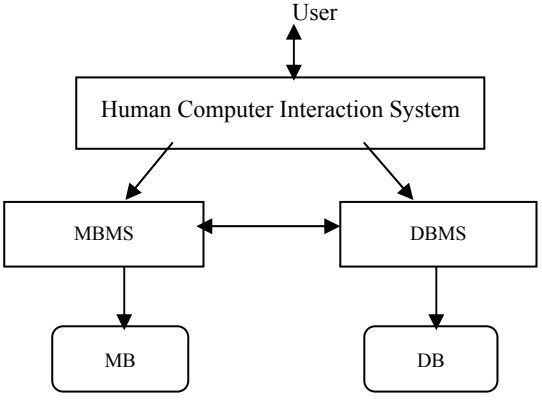


Figure 1 Traditional Structure of DSS

2. FEATURES OF DATA WAREHOUSE

A data warehouse is a subject-oriented, integrated, steady and time-variant collection of data, mainly applied to make decision. DW techniques gain original data from many traditional heterogeneous or congeneric databases, firstly form current basic data level according to assistant decision-making demands, and then form synthetical data level according to synthetical decision-making demands.

There are many ways to define DW, including the one [2] put forward by Inmon in 1992 who is regarded as "the father of Data Warehouse", which is

- Object-Oriented
- Integrated
- Time-Variant
- Nonvolatile

to support data integration disposed by decision-making.

Concrete characters are as follows:

(1)DW is Object-Oriented. In a DW, data are organized for primary objects (such as sale) of enterprise not for individual affair. In other words, this organization is based on topic fields of enterprise rather than on software application. The reason for such differences is that Internet applications are designed for procedures and functions, which require special data respectively. However, some data elements are only involved in certain part of functions. These operational data are concerned with prompt requirements of application and are based on modern business system. On the other hand, DW contains decision-making-oriented data, which might have existed for a long time and contained more complicated relation.

(2) DW is integrated. Data in DW derived from daily operational data are the increment and uniform disposal, such as uniform naming rules, uniform measurement units etc, rather than simple merger or move themselves because the structure and realization methods of daily operational data are various, which may adopt different code and different naming rule in order to pursue local optimization. As far as DW is concerned, how no matter daily operational data are designed and realized, the operational results must be consistent, in addition, data and methods must be saved according to single format, which is accepted in public. Only in this way, would the users of DSS not consider the problem of data consistency when they use them.

(3)Data in DW are non-updated. Data in DW are mainly provided to be used by decision-making analysis and data operations involved are mostly query and rarely modification. Data in DW reflect the content of history data over a considerably long period rather than the content of data processed on line.

(4)Data in DW are changed constantly. DW users don't update data when they are analyzing and operating. However, it doesn't mean that all the data always remain static in the whole existing cycle from the moment when people input the integrated data into DW to the moment when they are dropped form the DW lastly. As time goes on, DW needs to continuously add new data and delete old ones.

At present, we can make information and application shared via web; it has been the development trend of information technology to make original business increment by means of Web techniques. People are always seeking to get various decision-making information as much as possible and share various applications when they design DW. Thus, DW application based on Web has been the optimum scheme of constructing DSS presently.

3. DSS ARCHITECTURE BASEED ON WEB AND DW

With the rapid prevalence of computer net technology and the gradual development and maturity of DW techniques, On Line Analysis Processing (OLAP) and Data Mining techniques, the development of DSS has gained new chance. By virtue of the advantage of DW, new DSS makes up the deficiency of the old one and makes good use of the whole database resource in current system. Then, non-technology managers and normal users could access and analyze information derived from database conveniently and freely, and enterprise's work and analysis of DSS based on web and DW are supported, accordingly, the field and range of application of DSS are extended. Architecture of DSS based on web and DW is shown as figure 2.

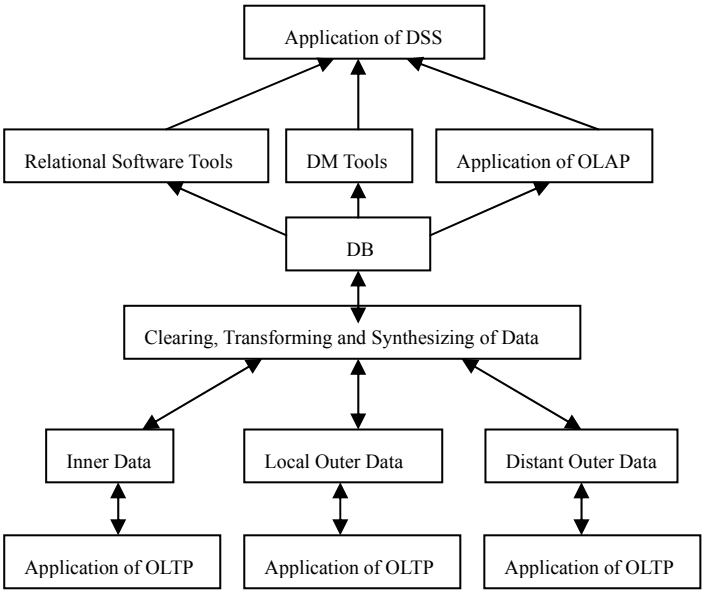


Figure 2 Architecture of DSS Based on Web and DW

Database in OLTP (On Line Transaction Processing) application system is the data source of the whole DSS and the material base of building DW. Data in DW are inputted, modified and deleted via data management program. Information derived from the outer is sent to DSS as the form of mail. Meanwhile, information in DSS is sent back to the outer as the same form. The outer terminals save the data into database for the future query when they receive mails. The clients of system send the login demand to the server with a simple browser (such as IE), then, the server check the users and dynamically return information requested to the permitted users when receiving requests. After logging in, the users can exercise their granted operation rights to DW and get data required respectively.

In this way, DSS based on Web and DW should adopt distributed Client/Server architecture, which is equipped with a Web server (NT Server4.0, IIS4.0 with Active Server Page), a DW server, a mail server and a dialing server. Web server is responsible for affording running environment for ASP program, managing system users, dealing with requests from users and communicating with DW server. DW server is mainly used to save information for future querying,

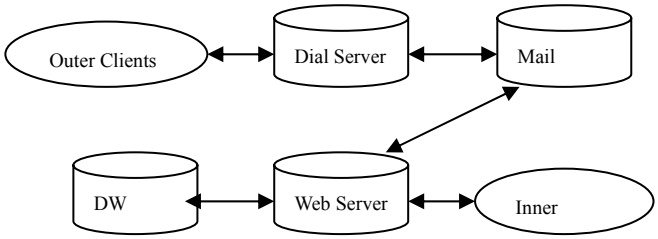


Figure 3 Client/Server Architecture

summarizing and saving. The function of mail server is to send and receive mails, keep in touch with the outer and get information. Dial server plays a role of channel for the outer users entering the inner net. The inner users may connect the server through local area net while the outer ones communicate with system by dialing phone numbers. Such architecture is shown in figure 3.

After integrating, converting and synthesizing the business-level data in OLAP database, DW re-combines them into public data view which is served as the base for data saving and organizing, and then, solves data inconsistency which occurred in the old DSS.

OLAP constructs analysis-oriented and multi-dimension data model according to the integrated data in DW, then, analyzes and compares the multi-dimension data from various aspects by multi-dimension analysis methods.

DM (Data Mining) automatically mines underlying data patterns from the mass data in DW and multi-dimension database and then automatically makes forecast from these patterns. Knowledge mined by DM may be straight applied to direct the analysis in OLAP while new knowledge from OLAP may be directly added into system knowledge base.

4. TECHNIQUES AND METHODS OF SYSTEM REALIZATION

At present, in most application system with Client/Server pattern, browser interacts with users only in the form of HTML and responds users' requests taking page as the unit, so, the performance of real time and appreciation is weaker. This DSS expands the function of client port by virtue of customized ActiveX controls and realizes efficient access to data by the method of gaining distant data and the ADO technology. At the same time, graphic operation interface and dynamic and efficient data communication system enormously improve the practicability of system.

4.1 Graphic method

In system, the function of real-time information processing is realized by virtue of customized ActiveX client control. We can create a ActiveX control and customize its attributes, methods and events and compile it into file taking .OCX as extended name in VB6.0, then, embed it into page in HTML and code script used by the server and client in which we have built attributes, methods and events in order to set communication among controls, clients and servers [3]. The

theory is illustrated as figure 4. When users send requests, browser is responsible for explaining the page passed by server. During the course, unregistered ActiveX controls would be unloaded and automatically register in local if they were found. Control instances are shown in browser in the form of visual graphic interface. Browser runs script and completes the initialization via communications between RDS (Remote Data Server) objects and database. Controls access database regularly and refresh the client with new data. When users' operations happened in browser, controls spring events whose codes would be executed and complete corresponding data disposal via RDS.

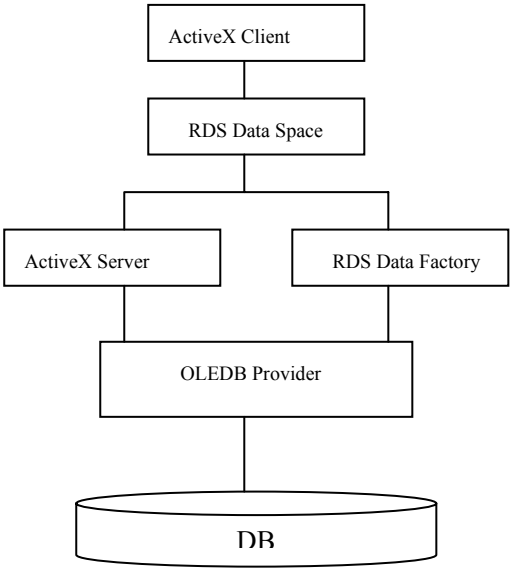


Figure 4 Communication Theories among ActiveX, Servers and DB

4.2 Method of gaining distant data

The interface named MAPI (Message Application Program Interface) is responsible for the realization of sending and receiving mails. MAPI, which consists of a serial of kernel system components, can connect all applications used in electric mail or workgroups with message server adapted to MAPI. Two steps of sending and receiving of mails are as follows:

(1)Logging in and setting up dialog with mail server by virtue of MAPI Session control. When it logs in, the control confirms electronic mail setting in operation system according to name and password provided by users and transfers basic message subsystem (electronic mail server).

(2)Managing personal receiving box by using all kinds of attributes and methods of MAPI control after dialog having been set up. For example, creating and sending message, adding attachment and verifying receivers' address in address book of electronic mail system etc.

Data pattern and mail topic are established in system and different types of data are sent as the different mail topics. After having received mails, system will trigger a data disposal message, which is responsible for ascertaining the source (from which department) and type (new data or modified data) of data by judging the address and topic of mails, reading mails data in fixed format and submitting to database after transforming properly, while taking out related daily information from database and sending mails back to the unit which sends data originally. After having completed, system retreats from dialog via MAPI and begins new next disposal.

4.3 Data access method

System adopts the scheme combining ASP with ADO. At server port, SQL commands in ASP files are executed to complete the access to database by virtue of ADO data object component. At client port, RDS is responsible for realizing data access function by virtue of script code of VBScript, JavaScript etc embedded into HTML pages. As soon as the server receives requests from the client, it will execute script commands in corresponding ASP files and generate standard HTML page returning to browser after gaining data required from database. Browser will execute corresponding script and complete page initialization or communicate with the server when it receives returned messages. Superiorities of adopting ASP and ADO techniques are as follows:

1) Script codes are executed at server port without support of users' browser;

2) Script codes are saved in HTML pages in the form of source code without compiling and linking, which makes development simple and prompt;

3) User obtain the running results of script rather than script codes themselves, which guarantee safety and secrecy of system;

4) Different results generated from the same ASP page may be returned according to different request, which improves interaction with users;

5) Combining with ADO objects makes Internet applications efficiently access information source of all kinds of database.

5. SUMMARY

At present, main researching productions in this field include efficiently improving respond speed of net, path pattern mining, intelligent query based on web (resource discovery, information extract, information summary etc), intelligent tools (intelligent search engine, intelligent browser, intelligent learning system, intelligent agent etc) and discovery arithmetic of user interest pattern etc [4].

DSS designed in this paper changes the traditional way of separately designing database, model base and method base and combines them into a multi-dimension database in which they are saved in the form of object and uniformly managed by database management system. Such DSS can preferably harmonizes the relation of data, model and method, thereby, the whole system is combined into a whole organic one which improve the integrity of system. However, there are still a lot of work to do in order to perfect the system, such as multi-ways of collecting information from net, making decision prompt and accurate and so on.

6. REFERENCES

[1] Chen Wenwei. Decision Support System and its development (the second version). Beijing: Tsinghua University Publish House, 2000,91.

[2] Chen Shuoying etc. Decision Support System in Data Warehouse. Beijing: Beijing Institute of technology Publish House, 2001,3

[3] Yang Jiahai etc. "Design and Realization of Distributed Network Real-time Supervising System Based on Web". Journal of Software,1999,10 (4):45~48

[4] Cheung D W. Discovering User Access Patterns on the World Wide Web. Knowledge Based on Systems.

Journal Elservier Science, 1998,10

Reconstructing Loop Space Technology for Parallel Loop and Realization in p-HPF Compiler *

Xuehai Hong , Qijun Huang ,Zhuoqun XU , Wenkui Ding
Department of Computer Science and Technology , Peking University
100871/Beijing , China
E-mail: hxh@ailab.pku.edu.cn

ABSTRACT

Parallel loop is one of the most important statements in HPF(High performance Fortran) , it's always used to express the core of most advanced science applications, such as oil & gas seismic exploration, weather prediction, N-body simulation ,and Gravitational Wave Extraction (GWE) simulation ,etc. The designing and implementation of parallel loop become a key aspect in developing HPF compilers. How to reconstruct loop space for loop parallel and realize it in p-HPF compiler becomes a very important problem, etc. The realizing technologies in p-HPF compiler were discussed in this paper including constructing the uniform partition sets, the technology of array index generating, the loop space reconstructing, communication detecting and organizing, etc.

Keywords: HPF language, Parallel loop, Parallel compilation, Parallel computation, Node program, Loop Space.

1. INTRODUCTION

High Performance Fortran (HPF) type of language is the object of our data parallel compilation. HPF is an extension to Fortran language, designed to deliver higher performance computation in scientific and engineering computing than what ordinary Fortran can do with automatic paralleling compilers.

HPF 1.0 specification was released May ,1993, which is mainly based on ISO Fortran language standard (Fortran 90) issued in 1992. It should be noted that both Fortran standard and HPF were evolving – after HPF1.0 , ISO is released Fortran 95 standard , which incorporates some of the HPF features ;HPF 2.0 δ was released in October,1996. For more information, visit http://www.crpc.rice.edu/HPFF/home.html. It is the newest standard High Performance Fortran (HPF2.0) .

The character of data parallel is the same operation for a larger number of data. This makes it very suitable using SPMD (Single Program Multi Data) node programs. p-HPF uses this way to compile HPF source programs to f77+runtime supporting library, and then use local compiler to compile the local node program into executive program. But there are some key problems during compiling loop statements in parallel compiler, that are as follow:

Computation partitions (CP): we usually use the who-own-who-processing rules to deal with simple computation partitions. But, in normal loop statements, there are many computing statements ,and there are many kind of statements in loop body, as well as there are different distributing ways for these arrays in computation. So we can't simply use who-own-who-processing rules to deal with computation partitions.

The transforming of global arrays to local arrays: Corresponding the global arrays to local arrays in CP, the visiting ways of global arrays are changed to the visiting ways of local arrays. Without special particularity, the name of global arrays should be the same as the local arrays normally. So the visiting ways of arrays are changed to the transforming of the arrays index. That is the transforming the global array index to local arrays index in HPF programs.

Inserting the communicating statements in node programs: The local node programs can only visiting local data, the visiting outside data only by communication. So the visiting programs receive the data, and the owning-data programs send the data during communication. All of this should be finished by cooperation among the different processors. This should be carry out by inserting communication statements (include receiving statements and sending statements) into the node programs with a whole communication layout.

Reconstructing loop space and eliminating the redundancies of the iterations. The change of the global loop space to local loop space is not only suitable for the node programs visiting local data, but also eliminates the redundancies of the iterations, and then improves the efficiency of the node programs running.

Moving the communication statements out and executing group communication: Moving the communication statements out from the loop statements can reduce the time that spend on the communication process in loop body and avoids the serious falling of the efficiency of program execution. After the work of moving the communication statements, the data which the whole loop visits and which isn't local could be sent during once group communication. The advantage all of this is, on one hand, that could augments the communicating data bag by collecting the scattered data bags, and then sends or receives it. This could reduces communicating times of visiting data, and improves the efficiency of program, on the other hand, that could shielding the communicating details of point to point, and makes the node program simple. Further more, we could optimize some kinds of models for group communication, and ulteriorly improve the efficiency.

Communicating detecting and communication organizing: During group communication, the communication models could be no-communication, SHIFT-communication and REMAP-communication. The communicating detecting makes sure that which model could be used. The communication organizing arranges the communication statements in node programs according to the communication model.

Dealing with correlative problems. In normal loop statements, there are many different types of statement and the visiting of different statements for the same array. This needs differently dealing with correlative problem for different visiting. For an example as following :

*Project 60173004 supported by National Natural Science Foundation of China.

```
!HPF$   INDEPENDENT
        DO   I=2, 99
!HPF$ INDEPENDENT
        DO J=3, 99
        A(I, J) = B(I+1, J-2)                    ①
        D(I-1, J-1) = E(I, J) + F(A(I, J) ), c)  ②
        END DO
```

Example 1 A Source HPF Program

Here, statement ① and ② visiting the same array A, so the node program must make ② visiting the array A in ① statement correctly while the value was updated.

All above are the key problems in compiling parallel loop statements. So we should discuss the computation partition (CP), and then deal with the correlative problems such as generating the local array index, reconstructing loop space and communication detecting and organizing, etc.

2. OPTIMIZATION FRAMEWORK FOR PARALLEL LOOP COMPILATIION

2.1 Parallel Loop Compilation

In the world, there are some HPF compilers for science and research. The famous compilers are such as dHPF of RICE University, SUIF of Stanford University, pHPF of IBM and pgHPF of PGI. The last two are for commerce. Comparing with our p-HPF compiler, they have their own characters. But there ere some differences between the famous compilers with our p-HPF, such as the strategy of computation partition (CP), the evaluation of axing parallel granularity, load balance, and communication among nodes, optimization of loop , etc.

The follow will discuss the compilation technologies for parallel loop statements in our p-HPF compiler. They are computation partition (CP), generating the local array index, reconstructing the loop space, communicating detecting and communication organizing and other correlative problems. The figure 1 is the flow chart of dealing with parallel loop statements in our p-HPF.

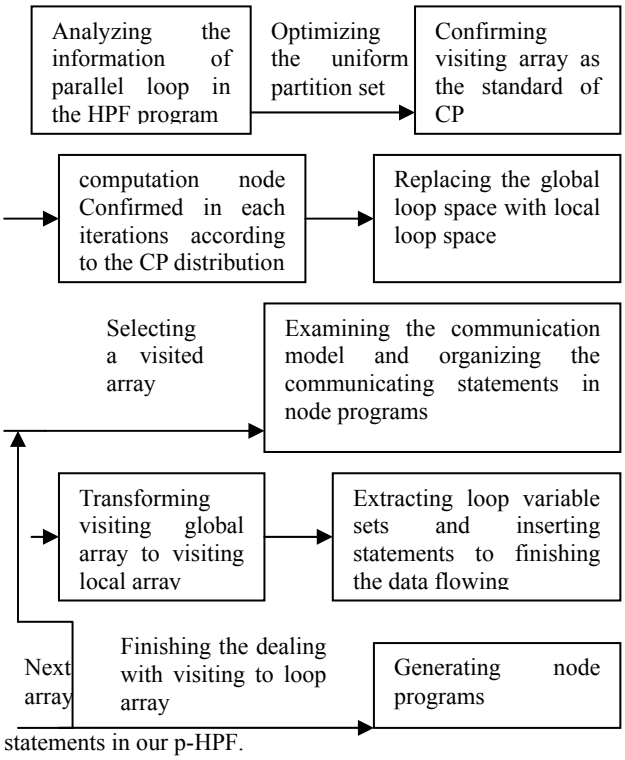


Fig. 1 The Flow Chart of Dealing with Parallel Loop Statements in Our p-HPF

2.2 Computation Partition (CP)

The who-own-who-processing rule is a simple computation partition. But we should not using this rule to deal with CP, because there are not only many computing statements in normal loop body, but also there are different statements, and different distributions in array in computation. So we use the rule of CP based on uniform sets. That can be understood as follow:

How to construct a scheme of computation partition sets (Cpsets), and how to evaluate each scheme in Cpsets, and then choosing the best scheme. The Cpsets should contain all kind of schemes theoretically. We can't ensure that the chosen scheme is the best one if the whole Cpsets could not be contained.

In order to keeping the regularization of visiting data, we usually don't do computation partition (CP) directly. We confirm the location of computation indirectly according to the location of arrays which are involved in computation, and then do CP. So we especially pay attention to the visiting to the array in the parallel loop body. The first example above, for that example, if we record the set as K which contains all visiting to array in the parallel loop body, then $K=\{A(I,J),B(I+1,J-2),D(I-1,J-1),E(I,J)\}$ The elements in set K could be recorded as *Ku.*

For this example, we have a scheme (recorded as CP0) which could be chosen:

The scheme of computation partition (CP) is made according to the array A(I,j) on the whole, that lets all the iterations of the number (I,J) taking place at one node where the element of array A(I,J) is there. In this case, the node which owns a piece of array A (2:99, 3:99) will share some of all iterations. The communications among the nodes which visit that array don't take place. But due to the different distribution between the arrays such as B, D and E with array A, the nodes owning arrays such as B, D, E, and A will communication when the nodes visit data each other, and this will result to spending during the communication. On other hand, due to the parallel program finishes only when all nodes programs finish the computation, so the peak value of node bearing the weight of computation directly impact the efficiency of the program running. Thus, the key evaluation for CP0 scheme is the peak value of node load and the spending of communication among the nodes.

2.2.1 constructing the uniform partition sets

If the set K is all array-visiting appearing in the parallel loop body, the whole set Cpsets of the partition scheme should be constructed by following steps.

For every statements Si such as ① and ② of the first example in one loop body, we temporarily confirm an element *Ku* in the ***K*** set, and then confirm the distribution of iterations of this statement *Ku* according to the data distribution of the statement *Ku* , so the *Ku* is the standard of the computation partition of the statement *Si*.. each statement *Si* is corresponding to the Ku standard of computation partition, $\{(Si,Ku),i=1,2,...n;u=1,2...m\}$,in this formula, the *n* is number of statements in the parallel loop body; *m* is the number of the elements in the set K. All the standard of computation partition for each statement is composed as meta-component. This becomes a CP scheme. If we change CP standard of any statement in the formula, the new CP scheme would be gotten. A CP scheme is one element of the Cpsets, all of this are composed as whole set Cpsets.

$CPsets=\{(Si,Ku),i=1,2,...n;u=1,2...m\}$.

The best scheme is selected from the CPsets.

2.2.2 The strategy of CP (Computation Partition)

In order to selecting the best CP scheme from the uniform partition CPsets, we could do it by the following three steps.

First , the distribution equilibrium of the uniform CP could be treated as the load balance, because the distribution of the visiting-array of the candidature CP among the processors reflects the equilibrium of the CP, and the different iterations of parallel loop could be treated as having the equal computing quantity in the rough. Secondly, for the every element CP in the CPsets, comparing other array-visiting with it, detecting the communicating need, and the communicating spending of the whole parallel loop could be evaluated from the communicating model of each visiting-array and the communicating quantity of data. The last, selecting the best CP from the CPsets according to the load balance and the communicating quantity of data for each CP. The third is very different, because the spending of the computing quantity of each iterations can't be expressed accurately in time during compilation, and the spending of communication is so. Thus the compare among the different CP couldn't be done.

In our p-HPF compiler, we make sure the load balance through the maximal number of iterations of the CP in one node. A candidature CP distributes definite number of loop iterations on every processor, and the maximal number of the iterations in all processors is the maximal iterations in one node. The more smaller the number of the iterations, the better the load balance. If the processors can't get the maximal number of the iterations in one node, we could get it with the number of 'span-processor'. The number of the span-processor is the processor where the visiting-array is distributed in.

In our p-HPF compiler, we make sure the spending of the communication with the communicating model and the number of the communication. There are three communication models such as non-communication, SHIFT communication and REMAP communication. From the angle of spending of communication, the non-communication is zero, the SHIFT one is little, but the REMAP is very much.

In our p-HPF compiler, we get the main factors and get the best CP by the consolidating step by step. We think the key factor of impacting the efficiency is the number of the REMAP communication according to a number of real testing, the next is the load balance of the CP, the last is the number of the SHIFT communication. According the order, an aspect is thought in each time, and then the CPsets is being reduced, at last, the best approximate CP will be gotten.

2.3 the Technology of Array Index Generating

HPF program visits directly any element in the global array using global named space, it is a global-visiting model; the compiled node programs only visit directly local array, it is a local-visiting model. So the compiler must solve the problem of transforming global-visiting model to local-visiting model, and transforming the global array index to local array index. Usually, even though the form of visiting to a normal array such as **A(l: u: s)** is regular in a HPF program, the form of local-visiting array is maybe not regular after the array mapped to node program, in this case, the local array index maybe be placed in a table, and the local array indexes are confirmed through the middle array. But the form of visiting-array is regular after mapped a piece of global array to the piece of local array in node, usually in the case of the HPF mapping. if only the local array index could be gotten and step, the local array index could be expressed simply, and the visiting model would be confirmed.

In the concrete, we use a compact way during transforming the global array to local array. In the node programs, the program call a runtime supporting function which computes the local array index such as subscript l_l, upper-scrip l_u and the step length l_s in node according to the distributed array descriptor (DAD) and the global array index such as subscript l, upper-scrip u and the step length s, and then the array index of every element which is visited in each iterations could be expressed by the l_l, l_u and l_s,. The follow is an example.

```
…..                          …….
!HPF INDEPENDENT             CALL local-addr(DAD_A,l,u,s,mp,ll, lu, ls)
  DO I=il, iu, is            do l= lil, liu, lis
  ……                         ……
  access(A(l:u:s))    ⇒      access(A(ll: lu: ls))
  …..                        …..
  END DO                     END DO
…….                          …….
```

Example 2 The Local Array Index Generating

2.4 the Loop Space Reconstructing

In example 2 , the whole global computing space ($\mathbf{i_l, i_u, i_s}$) of parallel loop on the left is transformed to local computing space($\mathbf{l_l, l_u, l_s}$) of node program on the right, this expresses accurately the iterations set on local processor., thus this is called the Loop Space Reconstructing, and eliminates the redundance of the iterations, and improves the efficiency of the node program. The loop space reconstructing is usually related to the partition and recombine of iterations space which is maybe very difficult. But all of that is avoided in our p-HPF compiler. The procedure is very simple and perspicuous. First we know the local loop space is computed according to the CP standard which is discussed above. Simply, the distributing location of the elements of a array which are visited in iterations decides the location where the iterations take place. Because the subscript of a piece of arrays which are distributed in any node is regular, (described above), and global array index usually is the linear function of the loop index variables, the loop index corresponding to the global index sequence is regular too. That is to say that the local loop index sequence could be expressed as (li_l: li_u: li_s). the example 3 is the follow.

```
…..                          …….
!HPF INDEPENDENT             CALL local-iter(DAD_CP, il,iu,is,a,b,lil,liu, lis)
  DO I=il, iu, is            do l= lil, liu, lis
  ……                         ……
  access(CP(a*I+b))   ⇒      access(CP(local-address))
  …..                        …..
  END DO                     END DO
…….                          …….
```

Example 3 Reconstructing the Loop Space

Supposed that the global computation (loop) space of a parallel loop statement is (1:100:1), the CP standard is A(I), the distribution of the array A is CYCLIC model, and the number is 2, then the procedure of reconstructing loop space can be pictured as follow.

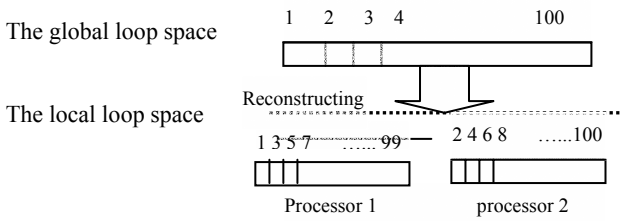


Fig. 2 The Example of Reconstructing Loop Space

2.5 Communication Detecting and Organizing

The node programs running on various processors visit non-local data through communications when they are running on the distributed memory and multi-processors structure system. The communicating statements are generated and distributed to suitable location in node program by the parallel compiler. This needs to compute the communicating set during compiling, and it is very difficult when the program could not make sure the communicating set or can't deal with the communicating set. For an example, if the bound of the array is variable, it is difficult for complier to organize the number of communicating statements such as "Recv" or "Send" in order to get higher efficiency. Besides, the node programs is very long , generating the code is complex, and the compiler is heavy-laden. In order to overcoming these disadvantages, we take the following methods and technology: The runtime supporting library ADLIB packs the bottom communicating statements such as "Recv" and "Send", and generates the communicating functions, and then give terse interface for the node programs. When the communicating function is called, the complex communication is finished by the communication functions only given the distributing information of the communicating array and the range of the index of the piece of communicating array as parameters. The advantages are as follow.

a. this strategy predigests the work of the compilation and the transform, and makes the node program codes terse.
b. Distinguishing the special communication model from the models, and optimizing the communication, and improving the efficiency of communication.
c. Organizing the communicating statements such as "Send" and "Recv" by manual way in communicating function, and improving the maximal efficiency of communication by using various optimizing technologies and strategies.

The model of the communicating functions is group communication formally. The communicating models in p-HPF are group communication such as non-communication, SHIFT communication and REMAP communication. In order to improving the efficiency of communication, we take different methods and technologies.

2.5.1 Communication Detecting

Communication detecting will solve the problems, such as which node program visiting non-local data, how distribution of these data, and how to get non-local data and update it while a communicating model (SHIFT or REMAP) will take. A normal form of communication detecting is in follow example. This form is common.

```
      type_x    X(Lx: Ux: Sx)
      type_y    Y(Ly: Uy: Sy)
   !HPF$  PROCESSORS  P(num)
   !HPF$  TEMPLATE  Tx(ltx: utx), Ty(lty: uty)
   !HPF$  ALIGN  X(l)  WITH  Tx(ax*I+bx)
   !HPF$  ALIGN  Y(l)  WITH  Ty(ay*I+by)
   !HPF$  DISTRIBUTE  (dist_type_x)  ONTO P::Tx
   !HPF$  DISTRIBUTE  (dist_type_y)  ONTO P::Ty
!HPF$  INDEPENDENT
        DO I=Li, Ui, Si          CP
        ……..
        access(X(I))
        …….
        access(Y(I))
        ……..
      END DO
```

Example 4 The loop of the CP confirming

Here, type_x, and type_y is any data type such as real; num is the number of logical processors; dist_type_x and dist_type_y are the type of data distribution, could be taken as BLOCK or CYCLIC. We confirm X(I) as the standard of CP, the communication model will be gotten by the communication detecting comparing Y(I) with X(I). The algorithm is that: ① enumerating the value of the loop variables I, computing respectively the processor mark where there are X(I) and Y(I) according to the distribution of X and Y, then judging the Y(I) is always in the place where X(I) is local for all I variables. If so ,then the communications for Y(I) don't take place, otherwise, go to ②. ②judging the conditions of communication model SHIFT for Y(I), if so ,Y(I) takes SHIFT communication., otherwise, go to ③. ③ Y(I) takes the REMAP communication model.

2.5.2 Communication Organizing

After confirming the communication model of array, the compiler organizing suitable communicating statements to finish the communication happened among the nodes in node program. It needn't communicating statements in the case of non-communication. The SHIFT and REMAP communication call corresponding communicating functions to finish communications respectively. The statements which communicating functions call are called communicating statements. In node program, the communicating function statements are placed outside of the loop body (reconstructing mentioned above). The data which computation need are obtained by placing communicating statements before the loop. However, the data which would be updated must be written back by placing communicating statements behind the loop.

3. CONCLUSIONS

The technologies for compiling parallel loop are discussed in this paper, we proposed a computation partition algorithm based on the subset of uniform schemes (uniform partition set), the method of getting uniform partition sets is given, As well as the algorithm of choosing the most optimized scheme under the consideration of communication and load balance. Then we discuss the technology in local array index generating, loop space reconstructing, communication detecting and organizing, etc. the high efficiency of this implementation has been proved by a lot of experiment. The p-HPF compiler adopting this compilation framework can obtain good speedups and efficiency.

The good works are as follow.

a. the uniform partition set is proposed on the basis of theoretic analysis and plenty of experiments. It contains the idea of distributing work with data which accords with the essence of data parallel. Based on the uniform partitioning ser, we guarantee the efficiency by the choosing the most optimized scheme.
b. The problem of statement-cost evaluating and comparing in compiling time is resolved in our scheme-choosing method. Load balance and least-communication are vital to acquire speedups, so they are considered synthetically here, we approach the cheapest partition scheme step by step by seizing the most important factor one at a time.
c. We adopted the strategy of group communication. Communication schedule and optimization are done comprehensively in communication function. Which makes the communication efficiency high enough to meet the need of parallel computing. Otherwise, this strategy greatly simplifies compiling and code generation. It also makes node program very terse.

4. REFERENCES

[1] http://www.cs.rice.edu/~dsystem/dhpf/dhpf-overview-96/index.html
[2] Michael Wolf, High Performance Compilers for Parallel Computing, Addison-wesley published,1996.
[3] HPFBench: A High Performance Fortran Benchmark Suite, Harvard University, Rice university,November,19,1999.
[4] QiJun Huang, Parallel Loop and Optimization for the Compilation of Parallel Loop, doc. thesis, Peking University,2002.
[5] High Performance Fortran Forum. High Performance Fortran Language Specification. Version 2.0, 1997. http://www.crpc.rice.edu/HPFF/home.html.
[6] J. M. Anderson and M. s. Lan. Global optimizations for parallelism and locality on scalable parallel machines. In Proceedings of the ACM SIGPLAN '93 Conference on Programming Language Design and Implementation, pages 112-125,June 1993.
[7] V. Adve and J. Mellor-Crummey. Advanced Code Generation for High Performance Fortran. In Languages, Compilation Techniques and Run Time Systems for scalable Parallel Systems, Chapter 18, Lecture Notes in Computer Science Series, Springer Verlag,1997.
[8] Jesong-Si Kim; Dong-Soo Han, Chan-Su, Parallel Loop Transformation Technique for Efficient Race Detection, Parallel and Distributed System,2001. ICPADS 2001. Proceeding . Eighth International Conference on ,2001,Page(s):265-272.
[9] P. Sipkova,V. rezany, Parallel I/O Support for HPF on Cluster, Cluster Computing and the Grid, 2001, Proceedings. First IEEE/ACM International Symposium on,2001,Page(s):186-193.
[10] Message Passing Interface Forum, "MPI-2: Extension to Message Passing Interface", University of Tennessee,1997.
[11] Xiaoming Li (NPAC), James Cowie (Cooperate System Coperation) , "On the Design of Distributed Array Descriptor",1996.
[12] Xiaoming Li, "Efficient Compilation of Forall Statement With Runtime Support", Syracuse University,1996.
[13] Xiaoming Li, "Runtime Enviroment Specification", Syracuse University,1996.
[14] Xiaoming Li, "Runtime Oriented HPF Compilation", CRPC-TR97694.

Research on building a VOD system with MPEG4 Technology

ZhengXiang
Institute of Computer Science and Technology, Peking University,
BeiJing, 100871, P.R.China
E-mail: zhengx@icst.pku.edu.cn

ABSTRACT

VOD（Video-On-Demand）system is a multimedia system which can storage abundant movie programs and provide multimedia service for large numbers of users. There are many factors prevent VOD from being widely application. Bandwidth is limited due to the use of modems, transmission reliability is an issue, as packet loss may occur, the need of different QoS , how to control and realize interactive VOD is an important area.
MPEG-4 is a better tool in solving those problems and boosting up VOD's function. There are a number of possible delivery platforms that could be tied into this technology for transport MPEG4 data. Some of these platforms are based on DMIF (Delivery Multimedia Integration Framework).
The organization of this paper is as follows: First, some terms that are used throughout the MPEG4 specification as well as in this paper are briefly introduced. Then, we introduce some VOD products which designed with MPEG4 technology and can be used in Internet. Next, we describe the DMIF(Delivery Multimedia Integration Framework) Communication model and presents the implementation of an MPEG-4 Streaming System in building a VOD client/server system with the MPEG-4 demonstration software implementation (IM1). Finally, we introduce a model which supported by DMIF and give a configure method for both server and clients.

keywards: VOD, interactive media, MPEG4, client/server model, DMIF

1. INTRODUCTION

Video-on-demand (VOD) [1]refers to video services in which users can request any video program from a server at any time. VOD has important applications in entertainment, education, information, and advertising. In fact, those areas ,VOD has been used in, include movie-on-demand, interactive video game, interactive news television, interactive shopping, interactive advertising, long distance learning , ITV （Interactive TV ,N-VOD，T-VOD）and so on. In some papers, MOD(Multimedia-On-Demand) is used as a typical example of VOD's application.
Due to its real-time nature, video streaming typically has bandwidth, delay and loss requirements. However, the current best-effort Internet does not offer any quality of service (QoS)[2]guarantees to VOD over the Internet. In addition, on the other hand, it is difficult to efficiently support VOD while providing service flexibility to meet a wide range of QoS requirements from the users. In order to address these challenges, and provide VOD services accommodating a large number of video titles and concurrent users, a VOD system has to be interactive, controllable and scalable --- scalable in streaming capacity.

Recent advances in digital video and audio compressing technology and the rapid development of MPEG4 technology have made it feasible to build video-on-demand services over metropolitan-area networks. A large-scaled and interactive Video-on-demand(VOD)system comprises many elements for the provision of the complete service.

The MPEG-4 visual standard [3] is developed to provide users a new level of interaction with visual contents, and provides technologies to view, access and manipulate objects rather than pixels, with great error robustness at a large range of bit rates. Application areas range from digital television, streaming video, to mobile multimedia and games.
The next content focuses on introducing an appropriate VOD system with low cost, low complexity, and offering high level of service quality (in terms of, for example, interactive operation or user delay experienced or user loss rate) which is designed according with MPEG4 technology and controlled by DMIF[4].

2. MPEG4 FUNDAMENTAL CONCEPTS

Audio or visual entities that participate as individual elements in a scene are termed audio-visual objects. Such objects can be either natural or synthetic. Synthetic objects may be generated with the graphics and synthesized sound operations provided by BIFS[5].
BIFS (scene description language) is actually more than just a scene description language, in that it integrates both natural and synthetic objects in the same composition space. Some objects may therefore be fully described within the scene description stream itself.
Elementary stream refers to data that fully or partially contains the encoded representation of a single audio or visual object, scene description information, or control information. In other words, elementary streams are the conceptual delivery pipes of MPEG-4; they are mapped to actual delivery channels using mechanisms that are described in detail later on.
Object descriptors include the scene description, audio-visual objects data, as well as object descriptor streams themselves. The information contained will include the format of the data as well as indication of the resources necessary for decoding (e.g., profile/level indication). Alternate representations or scalable encodings using multiple streams for a single audio-visual object can also be signaled by the object descriptor.
Scene description and stream description are strictly separated in MPEG-4. In particular, the scene description contains no information about the streams that is needed to reconstruct a particular audio-visual object, whereas the stream description contains no information that relates to how an object is to be used within a scene.
Timing of streams is expressed in terms of decoding and composition time of individual access units within the stream. Access units are the smallest sets of data to which individual presentation time stamps can be assigned (e.g., a video object plane). The decoding time stamp indicates the point in time at which an access unit is removed from the decoding buffer, instantaneously decoded, and moved to the composition memory. The composition time stamp allows the separation of

decoding and composition times, to be used for example in the case of bi-directional prediction in visual streams.
The actual packaging of the elementary streams as defined by MPEG does not depend on a specific delivery technology. MPEG-4 defines a Sync Layer that just packetizes elementary streams in terms of access units (e.g., a frame of video or audio data) and adds a header with timing and other information. Time stamps are readings of an object time base (OTB) that is valid for an individual stream or a set of elementary streams. At least all the streams belonging to one audio-visual object have to follow the same OTB. Since the OTB in general is not a universal clock, object clock reference time stamps can be conveyed periodically with an elementary stream to make it known to the receiver. This is, in fact, done on the wrapper layer around elementary streams, called the sync layer (SL). This is done in a uniform manner for all different stream types in order to ease identification and processing of these fundamental entities in each stream. All further mappings of the streams to delivery protocols and their control are handled by the delivery layer and are to be defined outside the MPEG-4 Systems standard. For example, transport of MPEG-4 content over RTP is to be defined under the auspices of IETF.
Furthermore, MPEG-4 has to operate both at very low and rather high bitrates. This has led to a flexible design of the sync layer elements, making it possible to encode time stamps of configurable size and resolution, as required in a specific content or application scenario. The flexibility is made possible by means of a descriptor that is conveyed as part of the elementary stream descriptor that summarizes the roperties of each (SL-packetized) elementary stream
Figure1 gives a VOD system which designed with MPEG4 technology and played by Realplayer player with envivio plug-in which used for supporting MP4format.
The mid-Autumn VOD system consists four movies whose content are relative with mid-Autumn Festival. While you select a movie from the list by pressing the left icon, you can watch the movie at the right area. The import & advanced factors are:
1. You can play some interactive action just as move, drag the play area and justify the appearance color which seldom used in an ordinary VOD system.
2. All VOD programs have been enclosed in one stream file which can be easy controlled by server
3. Developer can define each the VOD stream's sync function, such as bufferSize , timestamp, packetSeqNum's Length, AU seqNum's Length and other control messages in the stream file. Here, we give a sample control example:

```
ObjectDescriptorID 32
    EsDescr
    {
        ES_ID 2115
        OCR_ES_Id 2113
        muxInfo
        {   fileName "d:\\im1-2d\\debug\\4.263"
            streamFormat H263
        }
        decConfigDescr
        {
            streamType 4            // VisualStream
            objectTypeIndication  0xC2 // H263
            bufferSizeDB 16000
        }
        slConfigDescr
        {
            useAccessUnitStartFlag TRUE
            useAccessUnitEndFlag TRUE
            useRandomAccessPointFlag TRUE
            useTimeStampsFlag TRUE
            timeStampResolution 100
            timeStampLength 10
            packetSeqNumLength 3
            AU_seqNumLength 8
        }
    }
```

Figure1 A mid-Autumn VOD system

3. DMIF COMMUNICATION MODEL

MPEG-4 does not target any specific delivery technology but instead defines a framework which is referred to as the Delivery Multimedia Integration Framework (DMIF). It addresses the issues of local file access, broadcast media access, and peer-to-peer media access. Due to the exceeding demand in Internet multimedia, DMIF has sought to provide a flexible delivery platform for MPEG-4. DMIF has numerous advantages in that it abstracts the media from the delivery technology and enables easy utilization of various media ccess techniques. Other advantages of DMIF include the QoS provisions and the client/server capability exchange provisions, and so on.

Figure2 describes the position of DMIF used in MPEG4 system structure[6].

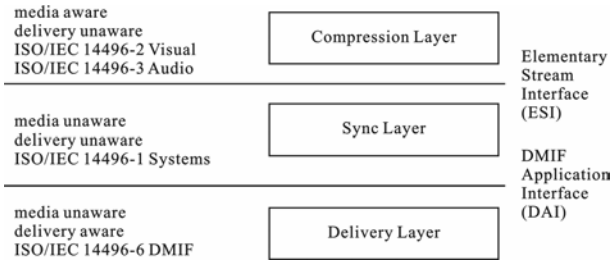


Figure2 The position of DMIF

DMIF has numerous advantages in that it abstracts the media from the delivery technology and enables easy utilization of various media access techniques.These advantages are evident through an implementation of an MPEG-4 streaming system that utilizes the DMIF as its media access framework. Other advantages of DMIF include the QoS provisions and the client/server capability exchange provisions.

The delivery efforts for MPEG-4 Version 2 will consider the following issues:

- Add mobile operation with DMIF.
- Extend DMIF V1 QoS to Access Unit Loss and Delay parameters at the DMIF Application Interface (DAI).
- Invoke session and resource management (SRM) on demand after an initial session has been established (with the tools present in DMIF v.1).
- Allow heterogeneous connections with an end-to-end agreed-upon QoS level.
- Integrate with Internet Engineering Task Force (IETF) specified network servers.
- Provide fully symmetric consumer and producer operations within a single device.
- Enable end-to-end sessions across multiple network provider implementations.

During VOD system, two communication planes are required: a Data Plane for the transport of media data (e.g., video stream), and a Control Plane used for media session management, and the Control and Data Planes can use different transport protocols. To ensure the reliability of Control Plane messaging in error prone environments, an error-free transport scheme should be employed. The architecture used for realizing the DMIF corresponds to the recommendations made in part 6 of the MPEG-4 standard and is depicted in Figure 3. An overview of the major components of the client/server system and the messaging that takes place between distributed peers is also seen. It is possible to use any session layer protocol instead of the DMIF Control Plane. Real Time Streaming Protocol (RTSP) is a primary candidate for this. One of the characteristics of MPEG-4 media is that it could potentially be composed of a large number of streams. DMIF has been specifically designed to handle these situations, whereas other streaming control protocols, including RTSP, would have to be adapted and greatly extended for such scenarios. Using RTSP in conjunction with DMIF provides functionalities that are not explicitly defined in the MPEG-4 standard.

In order to run MPEG4 file on DMIF structure, both ApadanaServer[8] and IM1-2D[9] program with DMIF Remote Instance are included. Here, we simple introduce how to configure the ApadanaServer and client.

Figure3 DMIF communication model for building a VOD system

A MPEG4-VOD system architecture conforms to the recommendations made by the MPEG-4 standard. This results in the existence of a Data Plane and an out-of-band Control Plane. The following paragraphs give an overview of the operations taking place.

During the VOD system running, when the media request has been made by the source application (the client), the initial object descriptor (IOD) will be sent to the client. All messages are exchanged between the originating and target DMIF layers and are transparent to the originating application, IM1 in this case. They correspond to the functionality within DMIF layers.

Once the session initiation messaging is completed , the server will request the media from the elementary stream provider and send this data to the client through the Data Plane when it receives a request from client. The flow of media data through the Data Plane from the server to the client and is as follows: Elementary Stream provider, packetizer, underlying transmission layer (i.e., RTP, UDP), client input Transmux channels, depacketizer (Sync Layer), decoder at the client application for playback[7].

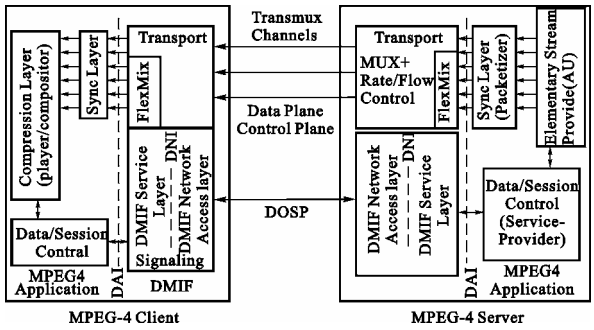


Figure 4 MPEG4-VOD System Architecture

ApadanaServer The ApadanaServer is an independent and stand-alone application and can serve any DMIF compliant client (e.g., IM1-2D). However the client is usually the IM1-2D program with DMIF Remote Instance included.

ApadanaServer uses a fast-start technique (and no start-up delay) to fill the jitter compensation buffer. To adjust the speed of the fast start scheme and therefore the size of the jitter buffer you should set the "client buffer size" to an appropriate value.by the technology, we can control the stream rate . For example: if the client implementation provides 10 seconds of buffering capacity, any value between the amount of jitter and 10000 ms will be acceptable for the "client buffer size". This value is pre-set to 1000 ms or 1 sec. If the quality of the streamed presentation is not satisfying, you might want to try other values.

The detail monitoring parameters include(Figure 5):

- Refresh Period (Sampling period or Observation Window in "seconds"): This parameter specifies how often the monitoring engine of the server should sample the outgoing bit rate. It also specifies the observation window for bit-rate measurement. By choosing small values for this parameter, it is possible to see the details of stream traffic. Using large values the average bitrate is measured.

- Maximum Displayed Bitrate: This parameter controls the vertical axis of the monitoring diagram, and has no effect on the measurement methods.

- Number of Samples Displayed: This parameter specifies how many samples of measured bit-rate should be displayed on the screen. Choosing a small number gives the ability to observe the traffic closely, a large number gives an overall view of traffic. This parameter however, only controls the horizontal axis and does not affect the measurement process.

When you adjusted the server (or if you are happy with the default values) you should invoke the monitoring process by pressing the "GO" button in the server window.

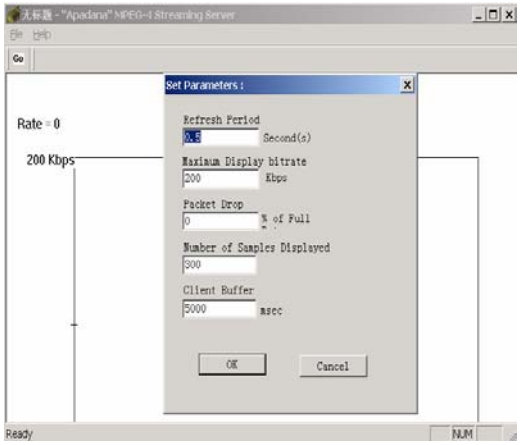


Figure 5 The configure of ApadanaServer's

Client To run the client, simply run the IM1-2D application. Start a presentation by choosing the "Open DMIF URL" from file menu. A dialog box will pop-up and you can enter the target URL in the designated area, this indicates the URL from which you want to receive the presentation. The URL format is specified in Annex C of the DM

<URL Scheme>://<username>:<password>@<Server IP address>:<Server DMIF Signalling port>/<filename>

For our case where the delivery platform is the Internet, two protocols for control plane signalling can be used, UDP or TCP. Both protocols are supported by the implementation of the Remote Instance. Therefore, the URL schemes in our case are "x-dudp" or "x-dtcp" respectively. The "x-" word in URL schemes stands for "experimental", it should be removed when the schemes are registered with Internet Corporation for Assigned Names and Numbers (ICANN).

The following gives an example of the DMIF URL:

x-dudp://zheng:password@192.168.2.14:5000/mymovie.mp4

Note that changing the URL scheme only affects the control plane, that is, the control plane messages are transported using UDP (x-dudp) or TCP (x-dtcp). Media data is transported in the data plane that uses a transport protocol irrespective of the Control plane choice.

4. CONCLUSION

VOD system is an importance system and is widely used around the whole world. In this paper, we gave some terms and a simple VOD system which relative with MPEG4 system and introduced a reference framework DMIF for supporting MPEG4 stream to run over Internet. Relay on DMIF framework, we can play an interactive VOD system with MPEG4 technology and control the play rate for difference network environment.

5. REFERENCES

[1] SW Carter, DD E Long and JF Paris. An efficient implementation of interactive video- on-demand, In Proceedings of the 8th International Symposium, IEEE, 2000, p172~179

[2] Y. Pourmohamadi. "Integration of Internet standard protocols and DMIF for QoS-aware delivery of MPEG-4 content over the interne", MASc thesis, Univ. of British Columbia 2001.

[3] MPEG-4 Overview - (V.21 – Jeju Version), ISO/IEC JTC1/SC29/WG11 N4668,March 2002

[4] Coding of Audio-Visual Objects – Part 6: Delivery Multimedia Integration Framework (DMIF), ISO/IEC 14496-6 International Standard, ISO/IEC JTC1/SC29/WG11 N2501, March 2000.

[5] Julien Signès . Yuval Fisher , Alexandros Eleftheriadis ,MPEG-4's Binary Format for Scene Description,

[6] http://leonardo.telecomitalialab.com/icjfiles/mpeg-4_si/5-BIFS_paper/5-BIFS_paper.htm

[7] G. Franceschini. "The Delivery Layer in MPEG-4," Signal Processing: Image Communication, vol.15, pp. 347-363.[7] , K. Asrar Haghighi, Y. Pourmohammadi, and H.M. Alnuweiri, Realizing MPEG-4 Streaming Over the Internet: A Client/Server Architecture using DMIF, April 2001, Presented at ITCC 2001

[8] Y. Pourmohammadi, K. Asrar Haghighi, A. Kaheel, H.M. Alnuweiri, S.T. Vuong. On the Design of a QoS-aware MPEG-4 Multimedia Server, International Symposium on telecommunications (IST2001). http://lan.ece.ubc.ca/QoSMPEG4_IST2001.pdf

[9] Kwang-Yong Kim, Hyun-Cheol Kim,Won-Sik Cheong, Kyuheon Kim. Design and Implementation of MPEG-4 Authoring Tool,www.cse.psu.edu/~cg585/paper/magecompression/mpeg4.pdf

Design and Test of MVICH ——Device Layer of MPICH for VIA

Haofei Liu
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, P.R.China
E-mail: lhf00@mails.tsinghua.edu.cn
Zhihui Du, Qunsheng Ma, Yu Chen, Chao Xie
Department of Computer Science and Technology, Tsinghua University
Beijing, 100084, P.R.China

ABSTRACT

The Virtual Interface Architecture (VIA), which defines a set of operations for data transmission, called Virtual Interface Provider Library (VIPL), is an industry standard for communication for cluster of computers. However, using VIPL to program is not an easy thing. So developing simple APIs over VIPL is required for feasibility of coding. MVICH is such a thing that, implements the functions of the Abstract Device Interface of MPICH (a portable MPI implementation) for VIA.
This paper presents design of MVICH first, introduces a variant modified for THVIA (a hardware-prototype of VIA) and then tests MVICH based on two VIA implementations: M-VIA and THVIA, comparing the results briefly.

Keywords: VIA, MPI, ADI, MPICH, MVICH

1. INTRODUCTION

The Virtual Interface Architecture Specification 1.0 [1] is proposed by Compaq, Intel and Microsoft, in Dec, 1997, to achieve low latency, high bandwidth communication network. Several VIA prototypes have been implemented, among which, M-VIA [2] and THVIA are involved in this paper. VIA brings great performance, but programming with VIPL is inconvenient in some degree. Researchers are seeking good solutions for this problem.
The Message Passing Interface (MPI) [3] is a standard of message-passing in the field of parallel programming. One of the implementations of MPI Libraries is MPICH [4], which, is widely accepted not only for its great applicability, but also its excellent portability. Accordingly, there is a natural idea that transplant MPICH from traditional network to VIA, to combine high convenience and high performance.
MVICH [5], supported by NERSC (National Energy Research Scientific Computing Center), is such an outcome, that just locates between MPI Library and VIA. This paper is concerned of the internal mechanism of MVICH and its variant for different VIA prototypes, and test results of both conformance and performance. The concept of VIA and MVICH is introduced briefly in the next part, after which comes a full-scale analysis of MVICH, with its modification for THVIA, and then the test results, at last this paper makes its conclusion.

2. OVERVIEW OF VIA AND MPICH

The Virtual Interface Architecture (VIA), proposed as an industry standard, is to provide low latency, high bandwidth communication for cluster of computers. VIA implements a network system that eliminates the processing overhead associated with the legacy network protocols, by providing user applications a protected and directly accessible network interface which, in the concept of VIA, is called the Virtual Interface (VI). Each VI is a communication endpoint, and no messages can be transferred before a connection between a pair of VIs is established. The VI architecture is composed of four major parts: the VI-NIC, Completion Queue, the VI Provider, and the VI Consumer. VIA supports two types of data moving: the traditional Send/Receive model and the Remote Direct Memory Access (RDMA) model, including RDMA Write and RDMA Read. User application is looked as a part the VI Consumer. Applications perform VI-pattern communication directly with the Virtual Interface Provider Library (VIPL).

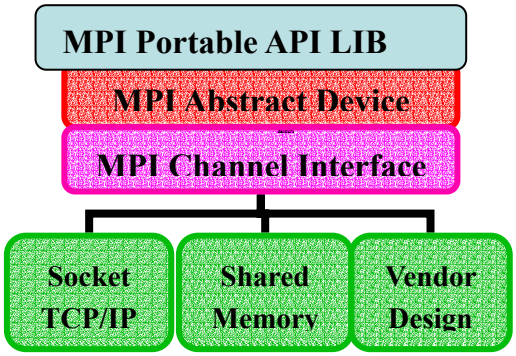


Figure 1 Structure of MPICH

MPICH is a complete implementation of MPI standard, and is also well-known for its portability. Figure 1 shows the structure of MPICH. The top layer is the library for user application call, and keeps unchanged no matter what the low-level device is. Next layer, called the Abstract Device Interface (ADI), is the most important part that provides portability. The Channel Interface is abstracted from ADI to simplify implementation of MPI devices. In fact, only five routines are required to be implemented to configure a device to Channel Interface. Another way is to implement a specific ADI directly. The ADI layer is between high-level MPI library and low-level MPI device, which has four sets of functions: ①specify send or receive of a message; ②data movement between API and device; ③manage lists of pending messages; ④provide information about execution environment. ADI defines interface prototypes for MPI Library to invoke, but interface routines are implemented separately for different

devices, since each device provides individual communication interface.

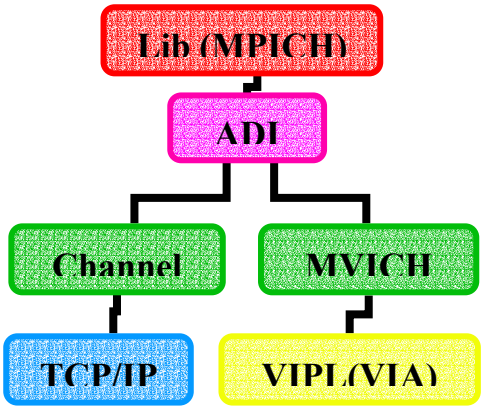


Figure 2 VIA to MPI

3. MVICH: DEVICE OF MPI

MVICH did not choose to support MPI via Channel Interface, so as not to compromise the system performance, and it just implements an ADI device for VIA. The relationship is showed in Figure 2. MVICH uses VIPL as its data transfer "device", which provides a high performance communication platform, instead of the operating system's network subsystem

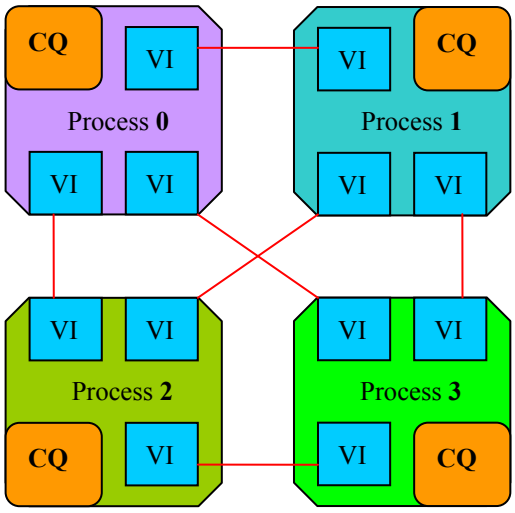


Figure 3 Connections of MVICH

using TCP/IP. The MVICH project comes together with the M-VIA project, both of which are developed by the same lab. M-VIA is a software prototype of VIA, while MVICH is to build MPI upon M-VIA; the former provides high performance, while the later provides applicability.

3.1 Analysis of MVICH

MVICH 1.0 is the latest release version, with the following features: "Credit" system flow control, multi-protocol, dynamic memory registration, VIPL peer-to-peer connection management, using of VIA completion queue.

As the "device" of MPI, MVICH firstly provides connectivity among all processes. Connection pairs of VIs are connected as in Figure 3. There are n (n=4 in Figure 3) processes, each has a connection to all other n-1 (3 in Figure 3) processes using VI, and all the VIs of one process are associated with a per-process completion queue, which is created before any VI can be created. The completion queue indicates a unique place of any completed descriptor in all VI work queues (two queues per VI), according to the completion time, and forms a sequence for the descriptors naturally. This time-based order will do great benefit to coding MVICH routines, since it ensures descriptors' processing order correspond to their completion order. However, this feature requires the VI provider to support the optional VIA component: CQ. By default, connections are established in a client-server manner, in which case, the lower-ranked process of each pair acts as the server. MVICH can also use the Peer-to-Peer model, if the VIPL implementation supports this kind of routines.

As MPI Library provides four protocols: Short, Eager, Rendezvous, and GET for data transfer, MVICH implements the following similar protocols to correspond:

- EAGER. For short messages, performs both of the Short and the Eager of MPI.
- R3. A standard three way handshake for normal messages, corresponding to the Rendezvous of MPI. It is a fallback of MVICH.
- RGET. Based on the RDMA Read capability of VIA, similar to the GET of MPI.
- RPUT. Based on the RDMA Write capability of VIA, symmetrical to RGET.

MVICH performs memory management with a special structure called Vbuf. A Vbuf is a registered memory region to VIA, and is a fix-sized buffer that is pinned in physical memory. Vbuf can carry control information of any protocol, as well as user data in EAGER and R3 protocol. All Vbufs and VI descriptors must be locked in memory during communication, and should be unlocked after that. As the execution is in system kernel level, the overhead is a little great, compared to other routines. A better scheme is to reuse, not only Vbufs, but also user-allocated buffers for RGET or RPUT. When freed, these user buffers are maintained by MVICH, not known to user, and are reused while new allocation is required. This is the so called Dynamic Memory Registration.

The flow control is present due to a fact, that any receiver of a connection pair must pre-post some Vbufs according to the VIA communication model. When the Vbufs are used up and none is freed or newly allocated, the transfer must suspend. The "Credit" system can ensure this. The sender is guaranteed some credits while establishing connection, each outgoing Vbuf consumes a credit, and no messages can be sent if no credits left. When the receiver has finished some items in the receive queue, it will pre-post freed Vbufs and piggyback a credit update to the sender which, then resumes the transfer. The receiver also sends "No-op" messages to the sender by interval, in case that no user messages are about to move in the reverse direction for a long time.

3.2 MVICH upon THVIA

THVIA is a hardware prototype of VIA, designed and developed completely by our lab, which belongs to the Department of Computer Science and Technology of Tsinghua University. Its VI-NIC is a PCI master controller, compatible of 66MHz 32-bit PCI Local Bus. Since THVIA is our first attempt to implement VIA based on hardware, its function is not complete. It provides 16 VIs, each with a hardware doorbell; user-level send operation, interrupt-based receive operation; peer-to-peer connection model; traditional TCP/IP transfer support. THVIA does not support user-level receive operation, RDMA operation, and the Completion Queue for the moment.

Some modification is needed to make MVICH properly run

for THVIA. Although MVICH only contacts with VIPL, which keeps unchanged for different implementations, transplantation from M-VIA to THVIA is not direct, because THVIA does not support all interfaces defined in VIA Specification 1.0. The main impact is from the lack of Completion Queue routines. This paper has mentioned the benefit that Completion Queue brings, but here we must cope with problems without Completion Queue. To express in short, the main solution is to replace properly the VipCQDone and VipCQWait VIPL calls with other provided interfaces, such as VipSendDone, VipSendWait, VipRecvDone, VipRecvWait, and something like that.

For VipCQDone, we use a loop, polling each VI's work queues in sequence, with VipRecvDone and VipSendDone accordingly. Noticing a little difference of VipCQDone from VipRecvDone and VipSendDone, it just indicates the current first completed descriptor without doing any processing, but the other two do. MVICH firstly invokes VipCQDone to return a completed descriptor indicating its work queue, then uses VipRecvDone or VipSendDone to do further disposal. So we temporarily add an interface to VIPL, called VipCheck, which can only query the completion status of each work queue. Figure 4 shows pseudo code of the substitution of VipCQDone. The loop invokes VipCheck for each VI's work queue first, once it returns a completed descriptor, the loop breaks, and corresponding VipRecvDone or VipSendDone is called for this descriptor.

```
1. for Process Rank = 0 to np-1 do
2.     if (Rank != myid)
3.     then
4.         Check the VI connected to Process
           Rank for completed descriptor,
           using VipCheck;
5.         if (VipCheck has found one)
6.         then Return this VI endif
7.     endif
8. endfor
```

Figure 4 Substitution of VipCQDone

For VipCQWait, we can not even use VipRecvDone and VipRecvWait, because they only care about one VI, not the whole. Fortunately, MVICH all invokes VipCQWait with parameter VIP_INFINITE, that is, don't return from waiting status until getting a completed descriptor. Current process is queued in system kernel, and wakes up upon a completion event. So we use similar loop scheme for VipCQWait, with some difference that, it the loops infinite, only breaks when VipCheck returns a completion status. This scheme is working at user-level, and will not give up processor until VipCheck success. So it is CPU-consuming, but has no impact to the local system, if only one process is running on each node. However, we can make the process sleep for a short time, e.g. 100 micro-seconds, at the end of each loop. Thus, it can save CPU cycles. Figure 5 shows pseudo code of the substitution of VipCQWait, where $\mathcal{F}$ notes the routines showed in figure 4.

```
1. Invoke F;
2. Check return code of F;
3. if (success) then return;
4.     else sleep a short time;
5. endif
6. goto 1;
```

Figure 5 Substitution of VipCQWait

4. TEST OF MVICH

We choose two types of test for MVICH: conformance test taken with MPICH's source and performance test of PingPong. Also, we test MVICH on both M-VIA and THVIA. Testing machines are two uni-processor IA-32 PCs, with 33MHz 32-bit PCI Local Bus, running two OS each: RedHat6.2 and RedHat7.2, both with original kernel. Two computers are connected by Intel Pro100B NIC with a CAT5 cable directly, and also by THVIA specific VI-NIC, Crossbar Switch and LVDS cables.

Table 1 Result of MVICH over M-VIA

PASSED	NOT **PASSED**
all tests in coll/ all tests in command/ all tests in context/ all tests in env/ all tests in profile/ all but 6 in pt2pt/ all tests in topol/	trunc, truncmult, cancel3, cancelibm, cancelmessages, self(suspend at 131072)

Table 2 Result of MVICH over THVIA

PASSED	NOT **PASSED**
all tests in coll/ all tests in command/ all tests in context/ all tests in env/ all tests in profile/ all but 12 in pt2pt/ all tests in topol/	trunc, truncmult, cancel3, cancelibm, cancelmessages, self(suspend at 131072) dataalign, isndrcv, reqfree, sendrecv, irsendinit, flood2

The first part is the MPICH conformance test. For M-VIA, it passes all but six tests, listed in Table 1. But for THVIA, there are as much as twelve tests that have not passed, which are showed in Table 2. This may result from both THVIA and rewriting of MVICH. More debugging is going on to point the cause of these problems. For this reason, performance test is rather rough. The testing case is a PingPong program. Compared to performance of both M-VIA and TH-VIA, the latency is not more 10 microseconds than that of VIA and the bandwidth is nearly 95% of that of VIA. We don't present numerical data here, because the results are not stable, that is,

getting different values at different runs. However, these results are enough to give a general picture of the system. We still need to tune MVICH at some degree, to produce more exact results.

5. CONCLUSION

MVICH implements a new ADI layer of MPICH, with VIA device, taking advantages of both VIA's speed and MPI's feasibility, thus provide a high performance and high applicable programming environment for parallel computing. Tests have illustrated that, MVICH is fairly good as expected. However, MVICH is not fully conformable to MPI Library yet, some tests are not passed up to now. For THVIA, there are still more. We just keep on working to find solutions as early as possible, to make it better to use.

6. REFERENCES

[1] Virtual Interface Architecture Specification 1.0, http://www.viarch.org.
[2] M-VIA: A High Performance Modular VIA for Linux, http://www.nersc.gov/research/FTG/via/index.html, 1999.
[3] The Message Passing Interface (MPI) standard, http://www-unix.mcs.anl.gov/mpi.
[4] MPICH-A Portable Implementation of MPI, http://www-unix.mcs.anl.gov/mpi/mpich.
[5] MVICH MPI for Virtual Interface Architecture, http://www.nersc.gov/research/FTG/mvich.

Buffer Overflow Attacks on Linux Principles Analyzing and Protection

Zhimin Gu
Department of Computer Science & Engineering, Beijing Institute of Technology,
Beijing, 100081, China
E-mail: zmgu@x263.net
And
Jiandong Yao
Department of Computer Science & Engineering, Beijing Institute of Technology,
Beijing, 100081, China
And
Jun Qin
Department of Computer Science & Engineering, Beijing Institute of Technology,
Beijing, 100081, China

ABSTRACT

In order to attack and obtain the remote root privilege, buffer overflow and *suid* program have become a common method for hackers. In this paper, the attacking principles using buffer overflow on Linux have been analyzed, and the corresponding protection strategies have been given too.
Keywords: buffer overflow, stack, EBP, libsafe

1. INTRODUCTION

With the increasing of accessing Internet, the network security problems have become the public's focus. Based on the safety report of CERT, buffer overflow attacks, which account for about half of all, have become the most common ones among all sorts of attacking methods. The C programming language has vulnerability, i.e., it does not automatically bounds-check array and pointer reference. By writing data whose length is more than the actual size of the buffer, the specific memory was modified. The attacking goal, changing the program executing sequence, was achieved. Buffer overflow attack exploits this very characteristic.

2. THE REASONS ABOUT BUFFER OVERFLOW

The programs written in C often were plagued with buffer overflows. Two main reasons were described below: 1.The C does not do bounds check for array and pointer reference. 2. The standard C library, such as *strcpy*、*strcat*, is not safe [1]. When a C program runs on Linux, the mapping of the process in memory was divided into three parts: code segment、data segment and stack segment. The stack segment was used for allocating space for auto variables and saving the arguments and return address when function calling occurs. The function calling was realized through pushing data into stack or popping data out from stack. So, when you writing data whose length is more than the size of buffer to the destination, the neighboring place will be overwritten, resulting into buffer overflow. Through changing the overflowing memory address, the hacker can get the logon privileges of remote system.
If the exploited program has suit bit, the hacker will get the root privilege.
So, bound-checking, should be done by the programmer, actually often be omitted, which results into the security problems to the applications written in C.

* The paper was supported by the fund of state scholarship 21307D05

3. THE PRINCIPLES ABOUT BUFFER OVERFLOW ATTACKS

3.1 C Function Calling

When function calling occurred, the saved data (including return address) on stack was called Stack Frame. A new stack frame was created for each function call. For the convenience of referring the local variables and arguments, many a CPU using a so called Frame Pointer (FP) or local basic register (LB) to point to one specific place of a stack frame [2]. For the CPU of Intel, BP (EBP) is used for achieving this function. When referring to the local variables or arguments, only positive or negative offset to EBP was enough. Because the stack increasing to lower address, the positive and negative offset to EBP is needed to access local variables and arguments, respectively.
The function calling in C is compiled to call instruction. Its executing procedure follows up:
1. The function entry arguments were pushed onto stack by opposite order;
2. Push IP (Instruction Pointer) onto stack (done by call instruction), i.e., the return address — RET;
3. Push the previous EBP onto stack so that it can be restored when the function exits. Let current EBP points this address;
4. Push the local variables onto stack;
5. Execute the function;
6. Leave the function (return).

When the function returns, copy EBP to SP, pop the stack, restore the former EBP、IP and clear the stack. The calling and exiting procedures are listed as below:

```
Calling:  pushl %ebp
          movl %esp %ebp
Return:   movl %ebp %esp
          popl %ebp
```

ENTER and LEAVE instructions of Intel can well be used for this purpose — the context saving when function calling and the context restoring when function exiting.

3.2 Buffer Overflow

Now let's explain how buffer overflow can happen by one example program.
One example is listed as follow:

```
function(char *buf_src)
{
```

```
        char buf_dest[16];
        strcpy(buf_dest, buf_src);
}
/* main function */
main()
{
        int i;
        char str[256];
        for(i=0; i<256; i++) str[i] = 'a';
        function(str);
}
```

We can see obviously from the program that the size of array *str* (256 bytes) exceeds greatly the length of *buf_dest* (16 bytes). Buffer overflow occurred.

The stack using conditions before and during the function calling were showed in the following figure:

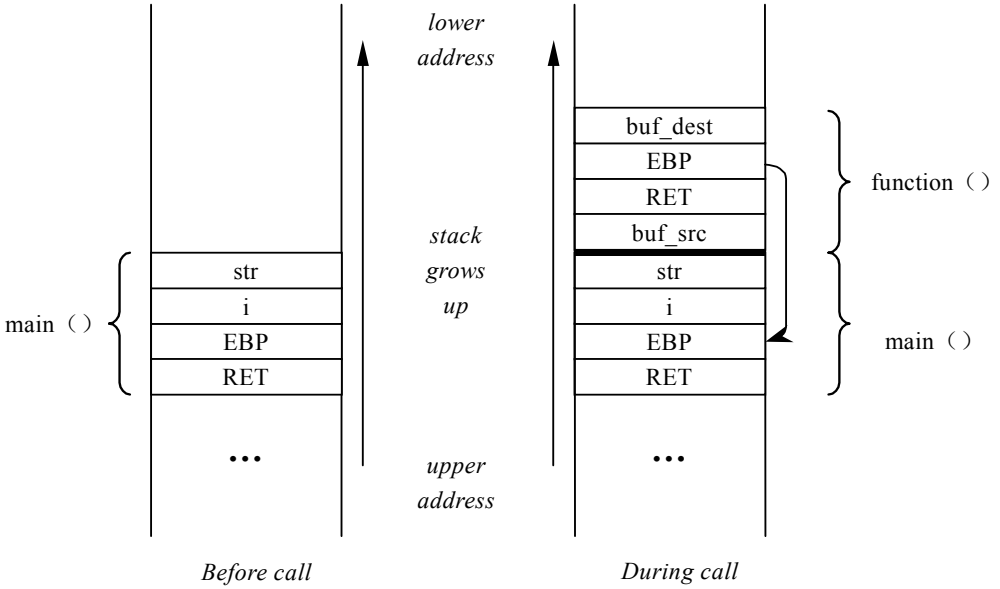


Fig. 1

We can see from this figure that the content of array *str*(256-character 'a's, i.e., 0x616161…) has already overwritten all the former contents from *buf_dest* to buf_dest+256, including the EBP and RET saved when function calling. So when the function exits, it will return to 0x61616161, the segment fault occurred.

Buffer overflow can change the executing sequence of program. If we can write a starting address of an elaborate attack code to RET, the system can be hijacked. If the attacked program has the *suid* bit and the attacking code obtained an interactive Shell, the hacker will get the root privilege. This is the main method used by hackers.

3.3 Shellcode

From the previous section, if you can give array *str* with specific content, you can let the program return to execute an special block of code so that you can attack the system. In order to get an interactive shell when the function returns, usually execute code: execve("/bin/sh", "/bin/sh",NULL). This code, having something to do with the specific assembler and machine hardware, was called *shellcode*. For example:

```
char shellcode[] =

"\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\xb0\x0b"

"\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
"\x80\xe8\xdc\xff\xff\xff/bin/sh";   [2]
```

This *shellcode* was acquired by disassembling. You can get one shell through overwriting only RET with the starting address of *shellcode*. The following program can achieve this function:

```
/* global variables */
char shellcode[] =    （see to the above discription）
char large_string[128];
/* main function */
void main() {
char buffer[96];
int i;
long *long_ptr = (long *) large_string;
for (i = 0; i < 32; i++)
*(long_ptr + i) = (int) buffer;        /* to stuff large_string with the beginning address */
/* of buffer */
for (i = 0; i < strlen(shellcode); i++)
large_string[i] = shellcode[i];    /* put shellcode to the foremost part of arge_string*/
strcpy(buffer，large_string);
}
```

In the above program, when returning from the function *strcpy*, its RET has already been modified as the beginning address of *large_string*, where *shellcode* lies. So *shellcode* will execute and an interactive shell will be obtained successfully. This was the result of executing *strcpy* without bound checking.

The above illustrates how to attack the program of our own. But in reality, we do not know the code of the attacking program, even the address of buffer. How can we attack an unknown program? It involves the construction of *large_string*. An effective method is to put *shellcode* at the middle of *large_string* and fill its former part with NOPs. It is then set it to an environment variable. By executing the program with the buffer overflow vulnerability using this environment variable as an argument, you can get a shell. Filling the former part with NOPs is for increasing the probability of returning to *shellcode* when exiting. If the return address points to any NOP, *shellcode* will be execute eventually, and shell will be got.

4. THE PROTECTION OF BUFFER OVERFLOW

The attacks using stack overflow listed above actually make use of the executable attribute of the stack, i.e., the codes in the stack can be executed. We can eliminate this trouble by making the stack non-executable. However, there are some fatal shortcomings of this method :

(1) In order to set the stack non-executable, everyone must make patch and recompile the kernel, but this is unpractical.

(2) *Nested function calls* and *trampoline functions* can be executed successfully depending on the executable attribute of stack.

(3) Setting the stack non-executable is unfit for another attack *return-into-libc* (pointing the program control flow to the shared library).

Libsafe 2.0, developed in Bell lab, can protect effectively the attacks by buffer overflow. The basic idea is: before calling the function with buffer overflow vulnerability, such as *strcpy*, intercepting this calling, check the distance from the destination memory address of writing to the places of RET and saved EBP in stack, comparing it with the length of writing content, if exceed, then overflow occurred, exit the function call.

Another method is avoiding the attacks using safe C library. For example, Richard Jones and Paul Kelly have developed *gcc/egcs* with bound checking, which can check most part of potential overflow problems . For windows, we can use the bound checking program made by NuMega. Its function is same to *gcc* with bound checking .

5. CONCLUSIONS

The C programming language, without the bound checking, has resulting in many programs with the problems of buffer overflow vulnerabilities. Hackers mainly overwrite the function return address by overflowing the variables in stack, and then coping a block of attack code to attain their aim. Libsafe can protect this kind of attacks effectively. We can also using the safe C library to programming, to make the programs free of buffer overflow vulnerabilities.

6. REFERRENCES

[1] Arash Baratloo, Timothy Tsai, and Navjot Singh. Libsafe: Protecting Critical Element of Stacks．White Paper．December 25, 1999

[2] Aleph One．Smashing The Stack For Fun And Profit ．BBS 水木清华站．Oct 1997

[3] Matt Conover (a.k.a. Shok) & w00w00 Security.w00w00 on heap Overflows. http://www.w00w00.org/files/articles/heaptut.txt．Jan uary 1999

Research on Ontology-Driven Product Data Management[1]

Hu Yujie
Institute of Artificial Intelligence, Zhejiang University
Hangzhou, Zhejiang, Post Code 310027, China
E-mail: blether@sina.com
And
Li Shanping
Institute of Artificial Intelligence, Zhejiang University
Hangzhou, Zhejiang, Post Code 310027, China
E-mail: shan@cs.zju.edu.cn
And
Yin Qiwei
Institute of Artificial Intelligence, Zhejiang University
Hangzhou, Zhejiang, Post Code 310027, China
E-mail: qiwei-yin@163.com

ABSTRACT

Increasing workforce diversity, globalization of markets and competition has driven the manufacturing to produce new and better products more quickly with much lower cost than ever before. There is an urgent demand for PDM systems to be much more efficient and powerful, which is leading to knowledge-based PDM systems. In this paper, on the basis of the discussion of the background and knowledge representation technology, especially focusing the field of ontology, a detailed description of the prototype ontology-driven PDM system that we are designing and implementing is put forward. As a result, conclusions are also presented.

Keywords: Ontology, PDM, Knowledge, DAML, RDF

1. INTRODUCTION

Increasing workforce diversity, globalization of markets and competition makes manufacturing around the world change rapidly. In response to new customer needs, competitive practices and emerging technologies, manufacturing science, information system and business practices must enable companies to produce new and better products more quickly with much lower cost than ever before. Information technology is the key to future manufacturing. During the past few decades information technology has made a significant impact in manufacturing and today it is a vital part of manufacturing by enabling manufacturers to efficiently generate and apply the vast amount of information that is needed to design, produce, and support their products. [1] But it is also driving manufacturing enterprises to delve ever more quickly into new realms of manufacturing technologies. Enterprises cannot afford to ignore information technology and the changes it will cause. Even more, information technology has changed the meaning of "manufacturing". Virtual Enterprise (VE) has cut across traditional corporate boundaries and is only a temporary consortium or alliance of companies formed to share costs, skills and exploit fast-changing market opportunities.[2]

When the marketplace continues to demand an ever-increasing level of product complexity, it becomes most critical to success in capturing, storing, and sharing manufacturing knowledge, including complete material knowledge bases, entire manufacturing process capabilities knowledge bases, and knowledge repositories of legacy products, processes, and people. In the major functions of a manufacturing enterprise information system: product design, manufacturing planning&execution and enterprise resource management, knowledge-based PDM system is the starting point and becoming an urgency because of enormous design activities and other planning. [3]

This article begins with an overview of the current underlying technologies of the PDM system, and some problems it encounters. Then the field of knowledge representation and ontology is introduced, which provides a good base for the design of a knowledge-based PDM system. Next, our knowledge-based PDM system that we are designing is put forward and implementing aspects are discussed in detail. Finally, conclusion and the future trends are presented.

2. BACKGROUND

Knowledge, however, is only valuable to a manufacturing enterprise if it is shared. In other words, knowledge must be explicit and " can be laid out in procedures, steps, and standards [4]." But it is a big challenge to share and exchange information in different format and form between many applications such as engineering design systems, process planning systems, enterprise resource planning systems etc.

In order to solve the product data sharing problems between heterogeneous systems many standards and languages have been put forward. Among them the most prevailing standard is STandard for the Exchange of Product data (STEP), which is a neural product model data exchange mechanism and capable of representing product definition data throughout the lifecycle of a product. And EXPRESS language is used to describe the information model of different domain. Under a uniform framework many application protocols (APs) have been developed as information exchange model to solve the data integration problem between different enterprise applications——specifically, application protocol 203. [5]

However there are still some problems of the current approach that should be given further considerations. First, APs are developed by different communities who come from various backgrounds. This allows people to give different understandings of the same concepts, even some of the

[1]The research is supported by National Natural Science Foundation of China (grant no 60174053).

common concepts. Although in each AP STEP has come to a consistent model of certain domain, things become much serious when integrating with the other APs. Secondly, even EXPRESS language and APs provide a powerful mechanism to describe information model, many systems have used other standards to model or wrap information in their own format. The information may be highly structured information such as Database schemas, EDI formats or weakly structured information such as text document web pages. The integration of the information is a non-trivial task in product data management.

Above all, current integration efforts are usually based solely on how information is represented (the syntax) without a description of what the information means (the semantic). And the knowledge in product development activities cannot be captured, stored and shared. Design database only contain geometry and documentation and much design knowledge is not stored and exchanged electronically.

We can see that many problems will be solved if knowledge or semantic meaning could be associated with the product data. Next-generation CAD/CAM/CAE tools will represent knowledge beyond traditional geometry, such as behavior (how the artifact achieves its function), form (the physical instantiation to achieve a function, including geometry, materials, assembly model, etc) etc. The goal of knowledge representation is to create data model or scheme to efficiently store, query, modify and reason with the information. Research in this field has given us a good starting point for the system design.

3. KNOWLEDGE REPRESENTATION AND ONTOLOGY

As a central-field of artificial intelligence, knowledge representation is a medium for pragmatically efficient computation by organizing information to achieve the recommended inference. [6] The goal of knowledge representation is to create schemes that allow information to be efficiently stored, modified, and reasoned with. Furthermore a knowledge representation scheme describes how a program can model what it knows about the world. Research in the field has also spawned a number of knowledge representation languages: frame-based, description logic, first (and second) order predicate calculus, object-oriented, etc. Every language is the result of the tradeoff between readability (how things are said), expressiveness (what can be said) and inference (what can be obtained from the information represented) in traditional and web-based ontology languages.[7] These languages differ in the way that knowledge is acquired, the extent of the descriptions they provide, and the type of inferences that they sanction.

Although knowledge representation formalisms support structures for organizing the knowledge of a specific domain, no mechanism have been provided for sharing and reuse it. In order to exchange information from different sources to be integrated, there needs to be a shared understanding of the relevant domain. Ontology, which is also an important terminology in diverse areas such as information retrieval, databases, knowledge management, and multi-agent systems, provides a common vocabulary to support the sharing and reuse of knowledge.

"*An ontology is an explicit specification of a conceptualization.*" [8] The conceptualization is the couching of knowledge about the world in terms of entities (things, the relationships they hold and the constraints between them). The specification is the representation of this conceptualization in a concrete form. So an ontology is a consensual, shared, and formal description of the important concepts in a given domain. Typically, an ontology contains a hierarchical description of important concepts in a domain, and describes crucial properties of each concept through an attribute-value mechanism. Additionally, further relations between concepts may be described through additional logical sentences. Finally, individuals in the domain of interest are assigned to one or more concepts in order to give them their proper type. So by defining shared and common domain theories, ontologies help people and machines to communicate concisely—supporting semantics exchange, not just syntax. Higher level built on top of it (with complex logics and the exchange of proofs to establish trust relations) will enable even more powerful functionality.

One challenging issue in supporting semantic interoperability with ontology is how to relate and align separately developed ontologies to use them together: adaptation of existing ontologies or composition of new ontologies.[9] Goh[10] has identified three main causes for this kind of semantic heterogeneity:

1) *Confounding conflicts* occur when information items seem to have the same meaning, but differ in reality, e.g. due to different temporal contexts.

2) *Scaling conflicts* occur when different reference systems are used to measure a value. Examples are different currencies.

3) *Naming conflicts* occur when naming schemes of information differ significantly. A frequent phenomenon is the presence of homonyms and synonyms.

Now there are many methods for this "ontology integration": merging, aligning, and relating etc.

Another challenge is the acquisition and evolution of ontologies: the ability to manage ontology changes and their effects by creating and maintaining different variants of the ontology. [11]

Next section we focus on the ontology-driven product data management system that we are currently designing and implementing.

4. ONTOLOGY-DRIVEN PRODUCT DATE MANAGEMENT ARCHITECTURE AND IMPLEMENTATION

4.1 Product information model and ontologies

The essence of the ontology-driven PDM system is to provide a knowledge base to the integration of product design and other processes or planning into one common activity. So product information model should aim at creating computer-understandable product representations that capture various aspects of product data such as its geometry, engineering attributes, manufacturing data, or bill-of-materials data. During the last decade or so, the focus of *product modeling* has moved from relatively low-level representations of product geometry to higher-level issues such as preservation of design intent, capturing various viewpoints to product data such as those of design and manufacture, and life-cycle management of product data. To preserve design intent, product model community has investigated approaches such as variable and feature-based product models and design history oriented models that relate the incremental evolution of product data with an explicit design process model. To facilitate process integration many enterprise integration frameworks also address *process modeling* which should codify concepts and connotations for describing products, engineering tasks, organizations responsible for carrying out

the tasks, main events and checkpoints occurring during the development, and all types of relations, constraints, and axioms that may be applicable for the process entities. And the entire *enterprise modeling* must be based on "*formal and shared models of the shared processes and shared data of the co-operating partners* — in short, *shared ontologies"*.[12]
The ontologies must base on the analysis of typical manufacturing scenarios: the identification of appropriate manufacturing scenarios, extraction of concepts inherent to that scenario, grouping of concepts into appropriate categories, and development of inference question that are based on those concepts. After a broad and accurate survey of the existing Enterprise Ontology, we choose the TOVE (Toronto Virtual Enterprise) as the base to build the PDM ontologies. The goal of the TOVE project has been the creation of an integrated set of ontologies to support enterprise modeling. TOVE ontology defines concepts used in enterprise modeling as object classes and its micro-theory contains a set of necessary and sufficient axioms to define semantics of those concepts. TOVE ontology includes several parts, such as activity, state, time, resource, cost and quality, so it can be used to specify a set of activities and resources and time constraints of activity in a specific enterprise. [13,14]

4.2 Knowledge representation

We also adopt DAML (DARPA Agent Markup Language) + OIL (Now DAML means DAML+OIL if not specified) representation language to represent the PDM ontologies, which has been developed by a joint committee from the US and the European Union (IST) in the context of DAML. [15] DAML builds upon RDF and description logic basis to combine the best features of other Semantic Web languages, including RDF(S), SHOE, and OIL. It allows class expressions to be a single class, a list of instances that comprise a class, a property restriction, or a Boolean combination of class expressions. DAML also provides primitives for defining properties. It also provides a means for handling synonymous terms, and provides some primitive version information. DAML is well suited for representing properties and relations of those products parts.
Through its extendable nature, DAML is able to decentralize STEP and map from these ontologies to others for non-engineering oriented agents. The following is an example which we use DAML to describe a sample part definition for assembly:

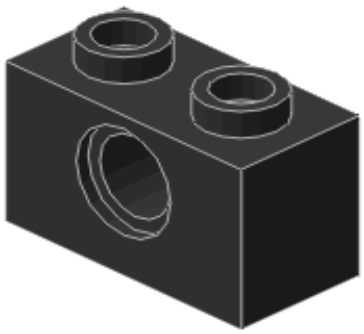


Figure 1 A sample part

```
<legos:TechnicBrick-1x2 rdf:ID="#RedBrick1" >
  <engpart:pose>
    <frames:Pose3D>
      <frames:position>
        <frames:Ground3DCoordinate>
          <frames:x>
            <frames:DistanceMeasure>
<frames:measure>
<legos:LDrawUnit rdf:value="0" />
</frames:measure>
            </frames:DistanceMeasure>
          </frames:x>
          <frames:y>
            <frames:DistanceMeasure>
<frames:measure>
<legos:LDrawUnit rdf:value="-48" />
</frames:measure>
            </frames:DistanceMeasure>
          </frames:y>
          <frames:z>
            <frames:DistanceMeasure>
<frames:measure>
<legos:LDrawUnit rdf:value="0" />
</frames:measure>
            </frames:DistanceMeasure>
          </frames:z>
        </frames:Ground3DCoordinate>
      </frames:position>
      <frames:orientation
rdf:resource="&frames;#OrientFront" />
    </frames:Pose3D>
  </engpart:pose>
</legos:TechnicBrick-1x2>
```

This logical approach to ontology design and implementation serves two purposes. First, it allows a precise and rigorous characterization of the consistency and completeness of the ontology with respect to its intended application. Secondly, it supports the implementation of automated inference for any enterprise model that uses the ontology. This second property is crucial in the work presented in this paper. Using the set of logical constraints, we were able to automatically infer that there were problems with the traditional PDM system within the enterprise model.

4.3 Ontology-driven PDM system architecture

In this section we describe the ontology-driven PDM system architecture that underlies our work in detail. We mainly focus on the architectural distinction between three main layers, namely the data, the management and the storage layer. Figure 2 depicts the layered architecture.

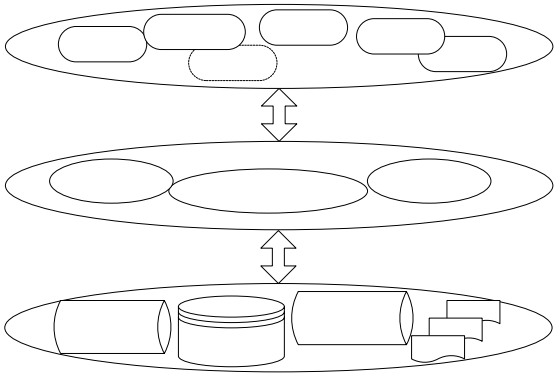


Figure 2 Ontology-driven PDM system architecture

4.3.1 Client Layer

Client layer consists of three types of applications: (1) flexible and extendable plug-in Integrated Developing Environment (IDE) , such as Onto-Edit, RDF-Edit or (2) applications extending the web-based framework or (3) other computer-aided systems such as CAD, CAE, CAM etc. And we can see that the integration of the PDM system with other engineering applications can occur at three levels: at the

start-up of any application connected with the PDM system, such as a word processor, CAD/CAM/CAE application etc; And an integrated interchange of attributes and/or metadata can also be automated between the applications and the PDM system.

4.3.2 Manager Layer

The management layer is mainly separated into two different parts: First, it is directly mapped onto the RDF-API which is an interface to the ontology and the knowledge base with standard methods to access the ontology. Secondly, it is mapped onto the PDM Server which is multiuser capable, allows persistent storage of models, uses transaction, security facilities and so on.

4.3.3 Storage Layer

The storage layer is also separated into four different parts: Ontologies Reprository, RDF Reprository, Tradition Database (Object-oriented Database etc) and Files System.

As ontologies are served as a consensual domain knowledge, their development require a cooperative process which has to be supported by central repositories that enable central storage and collaboration. So we are therefore currently implementing an ontology repository to leverage ontologies to the PDM system. This repository must keep up with the requirements as the traditional database including persistence, updation, concurrency, security etc. It is the same as the RDF repository.

However it is not possible for a single repository to provide the function for all possible representation vocabularies. First, there are various product data such as its geometry, engineering attributes, manufacturing data, or bill-of-materials data and the distinction of all the data is of very high importance in such an ontology-based modelling. From the following figure, in such a modelling a typical task is checking the consistency of ontologies. This involves structural and semantic checking of the conformance of a given set of relational metadata (RDF triples) towards an ontology as well as conformance of an ontology towards a given representation vocabulary. This task also poses special challenges to RDF repositories as a repository would have to understand the formal semantics of a given representation vocabulary. Secondly, scalability and performance of the system should be taken into account. Usually any ontology modification (comparable with schema modification in databases) can only happen when the part of the database affected by the change is not available to users. This is due to the fact that ontology (schema) modification normally implies data modification. Also, optimization techniques usually depend on schema and access structure.

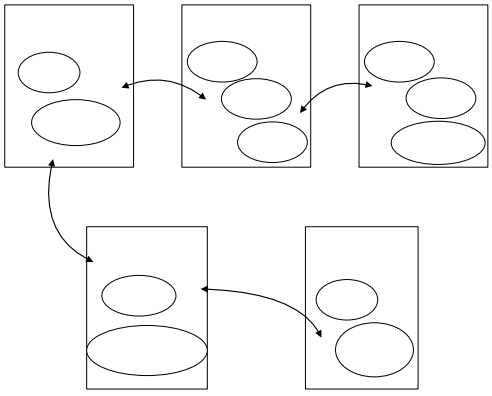


Figure 3 Representation Primitive Vocabularies, Ontologies and Instance [16]

Therefore, we propose to build a comprehensive infrastructure around a basic repository. We also believe that it is beneficial (especially for less tractable representation vocabularies) to use external inference engines Such as FaCT [17], to deploy a given RDF model whose ontology is fixed.

5 CONCLUSION

This paper describes an ontology-driven PDM system that we are designing and implementing. Compared with the traditional PDM system, this one has some remarkable characters:

- The system provides a shareable knowledge that stores, integrates and manages the various types of the product knowledge. It's important to represent the information such as requirements, versions, design rational, tasks, and main events with a common language for that it's the base to integrate without conflicts in the underlying semantics.
- Traditional Product data (such as CAD, CAPP) can be integrated seamless. This makes it possible to reuse the past product knowledge, experience and lessons to reduce the overall life cycle of the product.
- An excellent flexible and extensible system architecture is provided.

6 REFERENCE

[1] IMTR. Information Systems for Manufacturing Roadmap: Chapter 1, Introduction. July 24, 2000.

[2] Alexander V.Smirnov. Ontology-driven virtual production network configuration: a concept and constraint-object-oriented knowledge management. 2000.

[3] IMTR. Information Systems for Manufacturing Roadmap: Chapter 2, Product Design, Definition and Data Interchange. July 24, 2000.

[4] Dixon, Nancy M. Common Knowledge: How companies thrive by sharing what they know. Boston: Harvard Business School Press. 2000.

[5] Matthew West. An Overview of the Modularization, SC4 Framework, and SC4 Data Architecture Projects. ISO TC184/SC4/WG10/N291. 2000.

[6] Randall Davis, Howard Shrobe, and Peter Szolovits. What Is a KnowledgeRepresentation? AAAI, 1993.

[7] Corcho and Gomez-Perez. Evaluating knowledge representation and reasoning capabilities of ontology specification languages.2000.

[8] Thomas R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 1993.

[9] H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hubner.Ontology-Based Integration of Information —A Survey of Existing Approaches, 2000.

[10] Cheng Hian Goh. Representing and Reasoning about Semantic Conflicts in Heterogeneous Information Sources. Phd, MIT, 1997.

[11] Michel Klein and Dieter Fensel. Ontology versioning on the Semantic Web, Proceedings of the 1st Semantic Web Working Symposium, Stanford, CA, USA, July 30th-August 1st, 2001.

[12] M. Mäntylä and M. Ranta. Engineering Process Ontologies for Communication, Co-operation, and Coordination in a Virtual Enterprise, Helsinki University of Technology .1999.

[13] Gruninger, M., Atefi, K., and Fox, M.S., (2000), Ontologies to Support Process Integration in Enterprise Engineering, Computational and Mathematical Organization Theory, Vol. 6, No. 4, pp. 381-394.
[14] Fox, M.S., (1992).The TOVE Project: A Common-sense Model of the Enterpris, Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Belli, F. and Radermacher, F.J. (Eds.), Lecture Notes in Artificial Intelligence # 604, Berlin: Springer-Verlag, pp. 25-34.
[15] I. Horrocks and F. van Harmelen.Reference Description of the DAML+OIL Ontology Markup Language, draft report, 2001.
[16] Siegfried Handschuh, Alexander Maedche, Ljiljana Stojanovic and Raphael Volz. KAON — The KArlsruhe ONtology and Semantic WebInfrastructure, Institute AIFB, University of Karlsruhe. 2001.
[17] I. Horrocks. Using an expressive description logic: Fact or fiction? In Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98), Trento, Italy, June 2-5, 1998, pages 636–649. Morgan Kaufmann, 1998.

An Enterprise Manager for Clustered Database Servers: Managing All Services in One Suite *

Hui Liu
Software Institute, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
E-mail: linxin@xjtu.edu.cn
Junyi Shen
Software Institute, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
E-mail: jyshen@xjtu.edu.cn
Qinke Peng
Institute of System Engineering, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
E-mail: qkpeng@xjtu.edu.cn
Yi Xie
Institute of System Engineering, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
E-mail: xieyi@mail.cpc.xjtu.edu.cn
And
Cairong Yan
Software Institute, Xi'an Jiaotong University
Xi'an, Shaanxi 710049, China
E-mail: yancairong@mail.cpc.xjtu.edu.cn

ABSTRACT

In order to support dynamic web pages, a clustered web server (CWS) may be built on top of a clustered database server (CDS). For a convenient management of a cluster comprising a large number of nodes, it is necessary to develop an Enterprise Manager (EM) that can manage all services in a centralized way. Both demands depend on Management Replication. Focusing on the development of an EM for a CDS, this paper exposes the essential ideas, including concurrent control of management jobs, clone-able serialization of management jobs across all replicas, data replication scheme, fault tolerance, etc. The basic design requirements are discussed first. Then, main problems and their solutions, such as Active Nodes Discovery, Twin-keys Locking, Sequence Management and No-Acknowledgement-Strict-Order Publish-Propagate Replication scheme, are presented overall. Main data structures and system architecture are also described. Finally, some functionalities of a prototype EM are illustrated and the methods to encapsulate the content of a replication metadata are explained.

Keywords: Clustered Database Server, Clustered Web Server, Data Replication, Concurrent Control, and Serialization.

1. INTRODUCTION

Today, high performance web sites and E-commerce systems tend to rely on the web server farms [1] and clusters [2] for scalability, reliability, and low-latency access to Internet services. Both of them use interconnected PCs or workstations to build network supercomputers. We can regard a web server farm as an outspread cluster connected by Internet rather than LAN. Therefore, the differences between the systems reduce to one point.

The available research and commercial clustered web servers are ONE-IP, eNetwork Dispatcher, LSMAC, ACEdirector, Magicrouter, LocalDirector, LSNAT, LARD, IBM's Web Accelerator, ArrowPoint [3], etc. They focus mainly on the load balancing techniques. However, even the fastest clustered web server (CWS) faces two issues: the support of dynamic web pages and the management of a cluster comprising a large numbers of nodes.

For the first issue, our suggestion is to store all sensitive data into a database system, i.e., to build a clustered database server (CDS). For the second issue, the answer is to develop an Enterprise Manager (EM) that can manage web services, database services and other workstation basic services for the cluster in a centralized way. Fortunately, from the technical perspective, supporting dynamic web pages by a CDS and developing an EM for a cluster are almost the same problem. If we can dynamically manage a cluster and control its status, why dynamic web pages cannot?

As far as the design and implementation of an EM, this paper provides an exposure to concurrent control of management, ordinal execution of management jobs according to their dependency across all replicas, data replication scheme, fault tolerance, etc.

The rest of this paper is organized as follows. Section 2 summarizes the basic design requirements for an EM. Section 3 presents the main problems and their corresponding solutions. Besides divergence and intention violation, two other inconsistent problems for clone-able serialization, causality diversion and inverse causality diversion, are novel definitions. Four rules used to archive clone-able serialization are also given here. Section 4 introduces the main data structures and the system architecture of the prototype EM. Section 5 illustrates the prototype EM, showing how to replicate all kinds of files, operations and database updates to the selected cluster nodes in one suite. Section 6 concludes this paper.

2. DESIGN REQUIREMENTS

* This project is supported by national natural science foundation of China with the grant number of 60175015 and Xi'an Jiaotong University-IBM Bu'tone Software Center.

Dynamic web pages can be divided into transient type and variability type. Although transient web pages are produced dynamically, they never change the status of a web site. Quite differently, variability web pages update some information of a web site. For a clustered web server (CWS), the load balancing algorithms alone would make static web pages and transient web pages working well. However, a CWS must keep data consistency across all nodes to support variability web pages. One method is to store all sensitive data into some database systems. It means that a perfect CWS should be built on top of a clustered database server (CDS): a cluster equipped with database systems and servers as a unified database services provider.

There are three atomic architectures to build a CDS, sharing, partition and replication. In the sharing architecture, all cluster nodes access one shared database. In the partition architecture, each cluster node is equipped with a database system while the whole sensitive data are partitioned into slices and each slice is preserved in a given node. In the replication architecture, each cluster node is also installed with a database system. However, the sensitive data are replicated on each node.

The partition architecture is often used by the traditional distributed database systems. Oracle Parallel Server adopts the sharing architecture [4] and our CDS project employs the replication architecture. However, some common data needs to be replicated irrespective of the architecture being employed, such as cluster users, system time, etc. On the other hand, a cluster may comprise thousands of nodes. In this case, even a very little job, such as adding a new user, changing the time, shutting down parts of the cluster turns into a huge, repetitive, annoying task. Therefore, a software suite used to manage and monitor a CDS is very useful. In our project, we name such a suite as Enterprise Manager (EM). Of course, an EM may also be used to provide API to dynamic web pages.

The basic design requirements of an EM includes:

Defining selection set. Each management job may only affect the nodes defined in the selection set. In the replication architecture, the default selection set must include all replicas.

Managing files systems. Managing files systems of all selected nodes transparently. When we use an EM to create, delete, move, copy, paste, edit, deploy a file, all selected nodes must do the same job.

To execute shell scripts or batch files. Executing a shell script or a batch file on all selected nodes transparently. Almost all workstation services and system parameters can be managed with the shell scripts or batch files.

Services management. Managing the basic workstation services, such as web services, ftp services on all selected nodes transparently. Managing a service is always equal to editing some configure files and executing some scripts accordingly.

Monitoring cluster's status. Collecting and displaying the status information of all selected nodes is very important to system administrators of a CDS.

The design requirements above are general and easy to understand. For example, the management of files systems may include a file Explorer for cluster, a word processor for cluster, a Web composer for cluster, etc. In our opinion, we must try to summarize the requirements for an EM but not to enumerate them. Therefore, all EM requirements are condensed into **Management Replication**.

3. MAIN PROBLEMS AND THEIR SOLUTIONS

Management Replication intends to blend the distributed systems replication and the database replication together [5]. Because the EM users or dynamic web pages could issue management job asynchronously and symmetrically [6], Management Replication adopts an active, update everywhere policy [5] to implement relaxed consistency [1, 7]. Generally speaking, it includes three phases.

1) One node of a CDS issues a management job, executes it locally and encapsulates it into one replication metadata for propagation (We define this process as publishing).

2) The node that issued the management job propagates the replication metadata to all other replicas by some network protocol (We define this process as propagation).

3) All other replicas perform the management job wrapped in the replication metadata locally (We define this process as replication).

The issues on discovering and collecting useful information of all active cluster nodes, concurrent control of management jobs, management serialization across all replicas, data replication, etc., are discussed below.

Active Nodes Discovery

To discover and collect useful information of all active nodes composing a cluster is a fundamental problem of Management Replication. This information could be used to indicate which nodes are reachable, how much resources are used in each node, etc.

We use Active Nodes Discovery (AND) to find all active nodes of a CDS in a fault tolerance way. Two components, the Discovery Center and the Discovery Listener, are employed to achieve this aim.

The Discovery Listener is deployed on each node and is executed automatically at the start up time. It will register the local AND information into the Discovery Center when the node is started and deregister the local AND information from the Discovery Center when the node is shut down gracefully. Upon receiving a global AND information, the Discovery Listener will check its correctness with the local AND information and try to modify, re-register the local AND information when errors are detected. If have not receiving the global AND information or heartbeat for a given period of time, the Discovery Listener will restart the Discovery Center and restore the global AND information.

Only one active Discovery Center is needed in a CDS. It would multicast any changed global AND information to all nodes. In the meantime, it would send heartbeat to all known nodes periodically and try to eliminate all crashed or no-response nodes automatically.

Active Nodes Discovery could be very easily implemented with CORBA and/or Java RMI.

Concurrent Control

For a replicated CDS, each node may issue a management job to change its status concurrently. However, only one node could perform such a task at one time. Concurrent control of management jobs is an exclusive entry for all tasks that will change the status of a CDS.

Locking is a traditional and feasible technique for concurrent control. There are various kinds of terms that reflect different aspects of this technique. For example, read-lock, write-lock and site-lock focuses on its functionality; share lock and exclusive lock reflects the relationship between each lock-holder; compulsory lock (pessimism lock) and optimistic lock indicates when to perform concurrent control; two phase locking (2PL) and non two phase locking (N2PL) emphasizes how to acquire a group of related locks.

We introduce Twin-keys Locking (TKL) for concurrent control. It is named from the perspective of fault tolerance.

TKL allows either the lock-holder or the next applicant to unlock a locked target. Therefore, even a network failure occurred when the lock-holder tried to unlock a remote target, the next applicant would detect the error and unlock it. Because network failure is the primary errors occurred in a distributed system, TKL is especially suitable for the distributed systems.

In the normal situation, TKL must lock the target (the object that are to be changed) before updating and unlock it after updating. If the target had already been locked, TKL would query the lock-holder about its current status. New applicant will be rejected if the lock-holder is still using the lock. Otherwise, the lock will be revoked and granted to the new applicant if the lock-holder had finished using it or there is no response until time expired. Of course, TKL must be responsible for declining orphan unlock-requests and detecting certain lock conflicts.

The prototype EM possesses two kinds of TKL, the local lock and the global lease. The target on each node has a local lock. It is an exclusive entry for local updating and managing threads. The same targets on each replica share one global lease [8]. It is a mutex for all nodes that are going to update the target. When using TKL, neither the timer nor the renewal process is needed. Working together with the existed read-lock and write-lock mechanism in all kinds of applications, they could perform concurrent control for a CDS.

Clone-able Serialization

Concurrent control alone is not enough to keep data consistent for a CDS because its essential is to serialize concurrent management jobs. However, this kind of serialization only assures that the execution of concurrent management is correct on one site instead of across all nodes. For a replicated cluster, one challenge is to keep the result of management serialization identical or compatible across all replicas. We define it as clone-able serialization.

Because different serialization results of n management jobs could be n!, there are four potential inconsistent problems for clone-able serialization, i.e., divergence, intention violation, causal diversion and inverse causal diversion. The first two have been discussed in literature [9] together with two other inconsistent problems for CSCW system: causality violation and syntactic inconsistency. In our opinion, divergence and causality violation could be regarded as the same problem: serializing management jobs on a target in some different orders.

Denoting a target that makes reference to other targets at the updating time as a compound target, similarly, denoting a target that has been referenced by a compound target as a reference target, four potential inconsistent problems of clone-able serialization are defined below.

Divergence. Serializing the replication metadata on a target in different orders. For example, supposing the real serialization result is $O_{m,1}$, $O_{m,2}$, $O_{m,3}$, where m stands for the identity number of the target and 1~3 stands for the number of the serialization order, if the serialization result at another replica is $O_{m,1}$, $O_{m,3}$, $O_{m,2}$, divergence appears.

Intention violation. Publishing a replication metadata on a target according to stale serialization result instead of the latest serialization result. For example, supposing all replicas have the same copy of file whose content is "ABCDE" at time t_0. Then, node N_1 changes the file content to "ABECD" at time t_1. Unaware of this change, node N_2 wants to insert "F" after the first character "E" according to the stale serialization result "ABCDE" at time t_2, it intends to update the content to "ABCDEF". However, the final result may be "ABEFCD". It is quite different from the original intention of node N_2.

Causality diversion. Using stale serialization result instead of the proper serialization result on some reference targets. For example, there are two tables: T_1 has two fields ("No" and "Price") and its recorders are {{001, 50.00}, {002, 30.00}}; T_2 has two fields ("No" and "Stocks") and its recorders are {{001, 10}, {002,100}}. Supposing operation $O_{2,1}$ updates T_2 to {{001,45}, {002,10}}, which means current stocks of goods 001 and 002 are 45 and 10 respectively. Then, operation $O_{1,1}$ of "UPDATE T_1 SET Price = Price * 0.8 WHERE '20' > (SELECT Stocks FROM T_2 WHERE T_1.No = T_2.No);" at each replica may get different result depending on whether the replica had serialized $O_{2,1}$ before $O_{1,1}$ or not. If $O_{2,1}$ had been serialized before $O_{1,1}$, the result of T_1 is {{001, 50.00}, {002, 24.00}}. Otherwise, it is {{001, 40.00}, {002, 30 .00}}. Obviously, the former is right.

Inverse causality diversion. Using newer serialization result instead of the proper serialization result on some reference targets. Continue to think about the previous example, supposing operation $O_{2,2}$ updates T_2 to {{001,15}, {002,10}} after the execution of $O_{1,1}$, now, both $O_{2,1}$ and $O_{2,2}$ will influence the outcome of $O_{1,1}$. If the replica had serialized $O_{2,2}$ before $O_{1,1}$, another wrong result {{001, 40.00}, {002, 24.00}} would be obtained.

Sequence Management can be used to avoid four inconsistent problems above. It is based on three variables: global order number (GON), local order number (LON) and status order number (SON). GON shows how many replication metadata had been published on a target by all nodes. LON shows how many replication metadata had been executed locally on a target. SON indicates the exact status of a reference target when a replication metadata is published.

Encapsulating GON and SON information into a replication metadata and keeping LONs on each replica, Sequence Management obeys four rules bellow to avoid all inconsistent problems.

1) **Ordinal execution**. Executing a replication metadata only when **GON = LON + 1** is satisfied on the target. This rule is used to avoid divergence.

2) **Conditional updating**. A node must use the latest data to publish a replication metadata, i.e., **LON = GON** must be satisfied on the target before publishing a replication metadata. This rule is used to avoid intention violation.

3) **No stale data on all reference targets**. A node must use the latest data on all reference targets to publish a replication metadata on a compound target, i.e., **LON = GON** must be satisfied on all reference targets. On the other hand, the replication metadata on a compound target could be executed only when **LON ≥ SON** is satisfied on all reference targets. This rule is used to avoid causality diversion.

4) **Diversion transformation**. If a compound target had made reference to a reference target, we can conclude that the next update on the previous reference target would logically or really make reference to the previous compound target. For example, supposing the compound target is T_C and **GON = *m***, one reference target is T_R and **SON = *n***, according to this rule, there must be a logical or real update satisfying the following conditions. That is, the compound target is T_R and **GON = *n* + 1**, one reference target is T_C and **SON = *m***. Using this rule with rule 3 together could avoid the problem of inverse causality diversion.

By reduction to absurdity, we can prove that any serialization obeys all of above four rules is a clone-able serialization.

The prototype EM performs three kinds of Sequence Management. When a node is going to publish a replication metadata, a local Sequence Management must be performed first. When all clone-able serialization rules are satisfied locally and all local locks are obtained, the node would pass

the information of all targets (including both the compound target and the logical or real reference targets) and their LONs to a CDS agent to perform a global Sequence Management. Such information is defined as targets local status (TLS).
After receiving a TLS, the CDS agent will perform a global Sequence Management to get the information of all targets (including both the compound target and the logical or real reference targets) and their GONs. Such information is defined as targets global status (TGS). If all items of TLS equal the corresponding items of TGS, a replication metadata could be published. In all other situations, the node would not publish a replication metadata and some exceptions should be thrown.
Before executing a replication metadata, a replica will also perform a local Sequence Management first. When all clone-able serialization rules are satisfied locally and all local locks are obtained, the replication metadata could be executed. On the other hand, each successful update should result in a "diversion transformation", which will locally and globally introduce some logical reference relationships and eliminate some logical relationships that had been fulfilled.

Data Replication Scheme
In order to combine the distributed systems replication with the database replication, some problems must be settled are what kinds of data are to be replicated and how to replicate them?
In our scheme, two fundamental objects are used to encapsulate all kinds of data that are to be replicated. They are

File object. This object comprises all information related to a file, such as creator, owner, group, created time, modified time, security policies, full path name, content data, file type, etc. It can be used to replicate all kinds of files.

Job object. Job object derives directly from the File object. Because almost all operations could be written as shell scripts, SQL scripts or some similar batch files, we use Job object to encapsulate operations. Hereafter, we denote these two objects as replication metadata.
Because strict consistency is a wrong model for updating data [1], http benefits nothing from strict consistency and relaxed consistency improves performance by allowing clients to read stale data while some nodes are updating the CDS, relaxed consistency is employed for Management Replication.
Using IP Multicast instead of multiple point-to-point Unicast connections to replicate data reduces the network bandwidth required. Unfortunately, IP Multicast packets are not guaranteed to arrive intact at all destinations. So, replication scheme based on IP Multicast does not insure that the incoming replication data are complete. Although replication may benefits from reliable Multicast, it is not a necessary choice for replication because online-evolution and growth [10] weakens its effect and it is inconvenient for both parallelism and fault tolerance. Therefore, an error-retransmission mechanism should be introduced into IP Multicast to replicate the data in a fault tolerance way.
NASOPPR scheme is employed to replicate replication metadata in the prototype EM. It integrates the error-retransmission of IP Multicast, Twin-keys Locking and Sequence Management together to implement relaxed consistency. NASOPPR stands for No-Acknowledgement-Strict-Order Publish-Propagate Replication. The first two characters NA point out the characteristic of IP Multicast. The next two characters SO indicate the main features of Sequence Management. Publish-Propagate is something like but different from Publish-Subscribe [11]. The primary difference between them is that clients are voluntarily to be or not to be a consumer of the published information in a Publish-Subscribe system, while the information supplier selects the consumers forcibly in a Publish-Propagate system.
The error retransmission mechanism of NASOPPR scheme uses the difference between TLS and TGS to detect whether a node had lost some replication metadata and would awaken a retransmission process based on Unicast when necessary. The retransmission process will try to find the lost or fault replication metadata from its origin node first. If failed, the retransmission process will try to find them from some active nodes in turn. Thus, the lost or fault replication metadata could be recovered successfully, or, an exception of conflict will be thrown.
In fact, the error detection and data retransmission processes of NASOPPR scheme are postponed to the time when a node is going to publish a Replication Metadata. This can be thought as a "fat" Reliable Multicast.

4. DATA STRUCTURE AND SYSTEM ARCHIETURE

In the prototype EM, six main tables are used to implement NASOPPR scheme. Their definition are listed below, where the string before ":" stands for the name of the field, the string after ":" stands for the data type of the field and the field underlined constitutes the key of the table.

UDT. (target: String, gon: Long, type: Short, executed: Boolean, checkpoint: Timestamp, content: Object). It stands for Update Data Table. "Type" includes three kinds of values: file, sql and shell. "Executed" indicates whether the replication metadata has been executed or not. "Checkpoint" is the time when the replication metadata data finished execution. "Content" keeps the replication metadata as a special object. All published, received or executed replication metadata are kept in this table. It also can be used to provide the detail Log information.

LLT. (target: String, lon: Long, error_free_order_number: Long, locked: Boolean, locker: String, compound_target: String). It stands for Local Lock Table. "Error_free_order _number" is a number indicates that all replication metadata, whose GON is less than it, had been successfully received. "Locked" indicates whether the target has been locked or not. "Locker" is the ID of the current lock-holder. If the target is a reference target, "compound_target" indicates the name of the related compound target.

GLT. (target: String, gon: Long, locked: Boolean, locker: String, compound_target: String). It stands for Global Lock Table.

LDTT. (target: String, gon: Long, reference_target: String, son: Long). It stands for Local Diversion Transformation Table. It keeps all unfulfilled, local, logical reference relationships based on the rule of "diversion transformation".

GDTT. (target: String, gon: Long, reference_target: String, son: Long). It stands for Global Diversion Transformation Table. It keeps all unfulfilled, global, logical reference relationships based on the rule of "diversion transformation".

MIT. (target: String, gon: Long, contractor: String). It stands for Metadata Indexing Table. "Contractor" is the ID of the node from which the replication metadata originated.
The prototype EM consists of two parts. The first part is called Publish Center and the other part is called Contractor. There is only one Publish Center for each CDS while the Contractor should be deployed on every replica.
The system architecture of the prototype EM is shown in Fig.1. The relationships between main components are:

1) Some users manage a CDS through the EM GUI.
2) The Publisher communicates with the Discovery Listener

to define a selection set.

3) Communications of Active Nodes Discovery.

4) The Publisher applies for or releases the global leases that are used to control concurrent management jobs issued by different replica (Contractor).

5) The Lease Applicant produces some temporary tables to perform local Sequence Management.

6) Retrieving data from LLT, LDTT and UDT to refresh the temporary tables cached in memory.

7) The Publisher applies for or unlocks the local locks.

8) The Local Lock Manager produces some temporary tables to manage the local locks.

9) The Local Lock Manager changes the status of LLT to grant or revoke the local locks.

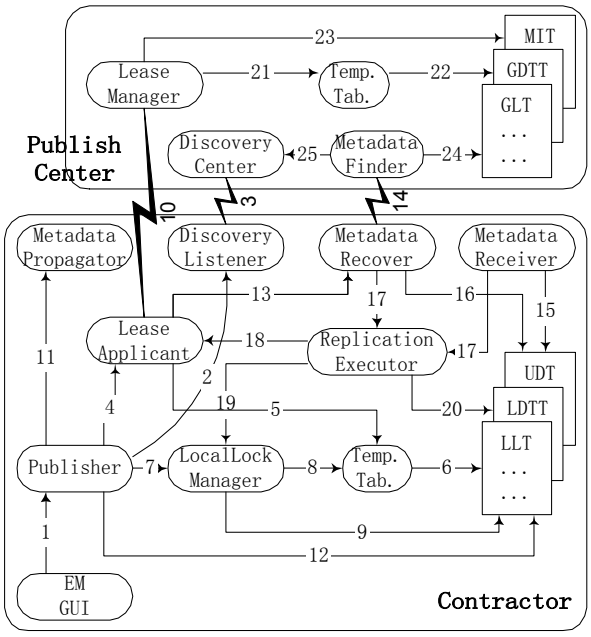


Fig. 1 System Architecture of the Enterprise Manager

10) The Lease Applicant communicates with the Publish Center to apply for or release the global leases.

11) The Publisher encapsulates the error-free and executed management jobs into a series of replication metadata for propagation.

12) The Publisher changes the status of LLT, LDTT and UDT after the propagation of replication metadata.

13) If necessary, the Lease Applicant would awaken the Metadata Recover.

14) Communications of Metadata Recover.

15) The Metadata Receiver changes the status of LLT and UDT after receiving a replication metadata.

16) The Metadata Recover changes the status of LLT and UDT after receiving a recovered replication metadata. Alternatively, it lookups a replication metadata in UDT for some Metadata recovering processes.

17) The Metadata Recover or the Metadata Receiver awakens the Replication Executor.

18) The Replication Executor asks the Lease Applicant to perform local Sequence Management.

19) The Replication Executor applies for or unlocks the local locks.

20) The Replication Executor changes the status of LLT, LDTT and UDT after executing a replication metadata.

21) The Lease Manager produces some temporary tables to perform global Sequence Management and global lease management.

22) Retrieving data from GLT, GDTT and MIT to refresh the temporary tables cached in memory.

23) The Lease Manager changes the status of GLT, GDTT and MIT to grant or revoke the global leases.

24) The Metadata Finder queries about the MIT to find the index information of a lost or fault replication metadata.

25) The Metadata Finder gets the information of all active replicas to find a lost or fault replication metadata.

In fact, Fig.1 only depicts the simplified system architecture of the prototype EM. In the real system, there are three kinds of dedicated components for the Publisher, Local Lock Manager, Lease Applicant, Replication Executor, Lease Manager and Temporary Tables. They are used for file replication, shell replication and sql replication respectively. However, all other components are shared for all kinds of Management Replication.

5. PROTOTYPE ENTERPRISE MANAGER

The prototype EM runs on top of Linux 7.1, Apache 1.3.12 and Oracle 8.1.6. Most parts of them are built with Java while some parts of them are written in shell scripts. Because beauty is in simplicity for the cluster software tools [12], we develop the prototype EM by means of integration. General speaking, the functionalities of the prototype EM are something like Explorer, JDBC Explorer, Oracle Enterprise Manager, etc. However, we improve their capability by allowing them to manage a CDS instead of a PC.

The prototype EM comprises four parts: database manager, file manager, service manager and cluster monitor. All parts uses the dialog box shown in Fig.2 to define the selection set.

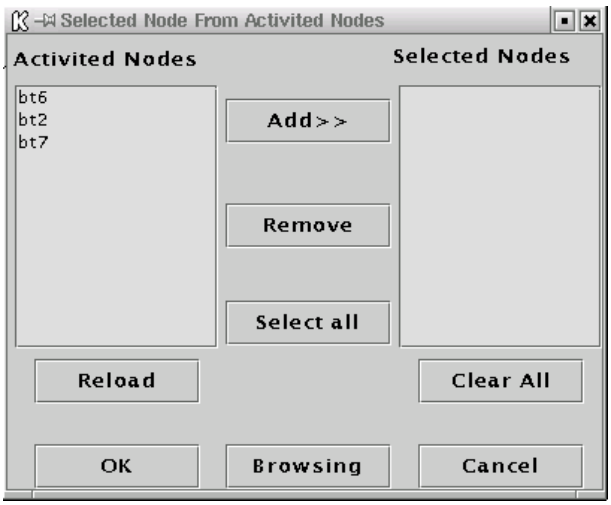


Fig. 2 Dialog Box of Active Nodes Discovery

Database Manager

Database manager is used to manage all database services in a CDS. It may include all kinds of database administration tools, such as schema manager, security manager, sql worksheet, etc. The prototype EM provides a schema manager, a security manager and a sql worksheet. The runtime schema manager is shown in Fig.3. Because it is not difficult to imagine its functionalities, we would not explain them in detail.

File Manager

File manager is used to manage files systems in a CDS. It can create, delete, move, copy, paste, edit and deploy a file in a CDS simultaneously.
If we want to create, edit and deploy a file with some special program, it also can benefit from file manager. We could first create and edit a temporary file with any program we appreciate, e.g., WinWord, Netscape Composer, etc. Then, we can use file manager to create a new file, specify its name and load the content from the temporal file. Finally, the file could be published with file manager and the temporary file should be deleted manually.
Because replication metadata uses byte arrays to convey the content of a file, file manager can be employed to deploy all kinds of files, such as jpeg, tar, gzip, rpm, etc.

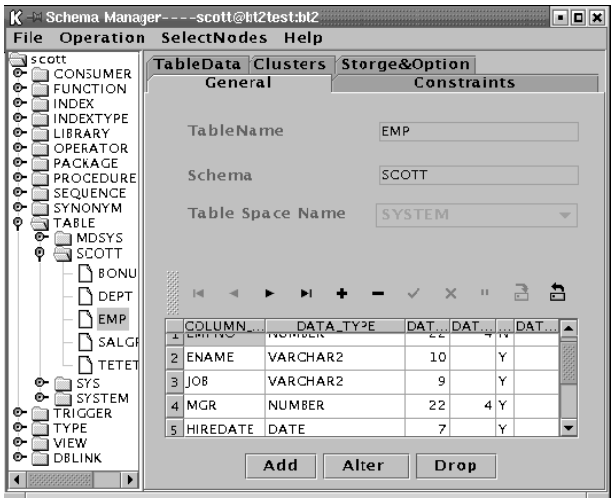


Fig. 3 Runtime Schema Manager

Service Manager
Service manager could be used to manage all kinds of basic services in a CDS, such as web service, ftp service, time service, sendmail service, reboot/shut down service, user/group information, etc.
In fact, all services could be managed in three fashions. In the first fashion, we can manage a service by executing some commands alone, e.g., time service, reboot and shut down service. In the second fashion, we can manage a service by editing its configure files alone, e.g., user and group information. In the third fashion, a service can be managed only by editing some configure files first and executing some commands thereon, e.g., web service, ftp service, etc.
For the first fashion, we can encapsulate all commands into a Job object. For the second fashion, we can publish a series of File objects. For the third fashion, we must encapsulate all commands and configure files into a Job object. The steps below assemble its content.
1) Setting all commands that should be executed before editing the configure files to the content.
2) Appending some commands that are used to parse, modify and save the formatted configure files to the content.
3) Appending all commands that should be executed after editing the configure files to the content.
4) Formatting the configure files into a single comment and appending the comment to the content.
Of course, the replication metadata can be handled as XML documents to improve its flexibility and variety.

Cluster Monitor
The most important status information includes information of CPU usage, disk usage, memory usage, system workload, active threads, etc.
Fortunately, Linux 7.1 provides many status guards that could be run on X Server. Because X Server can display the GUI of a remote X client, we can use these guards to monitor any selected cluster nodes conveniently. For example, Fig.4 shows a runtime xosview, which had been easily integrated by the prototype EM to monitor the CDS.

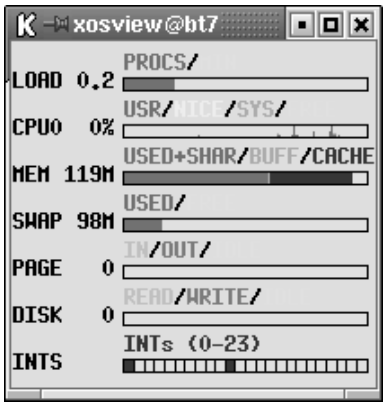


Fig. 4 Xosview Integrated

6. CONCLUSIONS

Network supercomputer constructed by the web server farms and clusters show great potential for high performance Internet services. We regard these two infrastructures as an analogical concept: let PCs or workstations to work together. This booming tendency also implies heavy research and industrial demands for clustered database servers.
Building a CWS on top of a CDS to support dynamic web pages and developing an EM to manage all services for a CDS in one suite meet the same problems: Management Replication. The challenges here are concurrent control of management jobs, clone-able serialization of management jobs, replication scheme, etc. By means of the prototype EM, the essential of all these challenges are exposed and some feasible solutions are provided.
Twin-keys Locking allows either the lock-holder or the next applicant to unlock a target, it is especially suitable for the distributed systems where network failure is the primary errors. Sequence Management obeys four rules to avoid all potential inconsistent problems for clone-able serialization, including divergence, intention violation, causal diversion and inverse causal diversion. NASOPPR scheme binds TKL, SM and IP Multicast together to form a very fat Reliable Multicast. This paper highlights main ideas, basic rules, core data structures and system architecture of these solutions.
Using these techniques, the prototype EM can replicate all kinds of files, operations and database updates to all selected nodes in a CDS. Therefore, it could be a very powerful administration suite and a very necessary middleware for clustered database servers.
In our opinion, the future work include
1) Providing a JSP API for web developers to use dynamic web pages in a CDS.
2) Detecting the replication metadata conflicts and resolving them automatically.
3) Parallel scheduling of the clone-able serialization results.
4) Handling the replication metadata by XML techniques.
5) Merging related management jobs warped in the replication metadata (Optimal scheduling).

All in all, how to design and implement a clustered database server is very interesting and is full of challenges and opportunities.

7. ACKNOWLEDGEMENT

We would like to thank the team of graduate students who have contributed to the implementation of the prototype Enterprise Manager Suite, including Xie yi, Wang muchun, Yan cairong, Wu hongjiang and Wang hua.

8. REFERENCES

[1] Randal C. Burns, Robert M. Rees and Darrell D.E. Long. “Efficient Data Distribution in a Web Server Farm”, IEEE Internet Computing, Vol.5, No.4, July/August 2001, pp.56~65.

[2] Mark Baker and Rajkumar Buyya. “Cluster Computing: The Commodity Supercomputer”, Software-Practice and Experience, Vol.29, No.6, 1999, pp.551~576.

[3] Trevor Schroeder, Steve Goddard, and Byrav Ramaurthy. “Scalable Web Server Clustering Technologies”, IEEE Network, Vol.14, No.3, May/June 2000, pp.38~45.

[4] Deborah Steiner, etc.. “Oracle Parallel Server Getting Started Release 8.0.5 for Windows NT”, Part No. A64425-01, Oracle Corporation, 1998, Chapter 1.

[5] M. Wiesmann, F. Pedone, A. Schiper, etc. “Understanding Replication in Databases and Distributed System”, 20th International Conference on Distributed Computing Systems, 2000, pp.464~474.

[6] Steve Bobrowski, Gordon Smith, etc. “Oracle8 Replication Release 8.0”, Part No. A58245-01, Oracle Corporation, 1997, Chapter 1.

[7] David B. Ingham, Santosh K. Shrivastava and Fabio Panzieri .“Constructing Dependable Web Services”, IEEE Internet Computing, Vol.4, No.1, January/ February 2000, pp.25~33.

[8] Randal C. Burns, Robert M. Rees and Darrell D.E. Long. “An analytical study of opportunistic lease renewal”, 21st IEEE International Conference on Distributed Computing Systems, 2001, pp.146~153.

[9] C.Sun, X. Jia, Y. Zhang, Y. Yang, and D. Chen. “Achieving convergence, causality preservation, and intention preservation in real-time cooperative editing systems”, ACM Transactions on Computer-human Interaction, March 1998, pp.63~108.

[10] Eric A. Brewer. “Lessons from Giant-Scale Services”, IEEE Internet Computing, Vol.5, No.4, July/August 2001, pp.46~55.

[11] Guruduth Banavar, Tushar Chandra, Bodhi Mukherjee, etc. “An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems”, 19th IEEE International Conference on Distributed Computing Systems, 1999, pp.262~272.

[12] Putchong Uthayopas. “Cluster Software Tools: Beauty is in Simplicity”, 2001 IEEE International Conference on Cluster Computing, 2001, pp.247~249.

Analysis of the Methods of Data Storage Using WWD & Cluster Software RAID

ZHENG shi-jue[1 2]
Department of Computer Science, Central China Normal University[1]
Wuhan, Hubei 430074, P.R.China
E-mail: zhengsj@ccnu.edu.cn
and
ZHANG jiang-ling
School of Computer Science and Technology, Huazhong University of Science and Technology 2
Wuhan, Hubei 430074, P.R.China
E-mail: jlzhang@hust.edu.cn

ABSTRACT

This paper, we discuss the Methods of Data Storage Using WWD & Cluster Software RAID, describe the WDD a large scale multi-wavelength network in essence can be considered like a large disk drive, with each wavelength being a separate track and specialized routers with the WDD software acting as record, analyze Data Storage using Cluster Software RAID, and, look forward application technology of WDD in the future.

Keywords: WDD, Cluster system, Data storage, Software RAID, Virtual Disk Drives, Virtual Router

1. INTRODUCTION

Wavelength Disk Drive (WDD) is a research CA*net 3 project at CANARIEInc since 1998[1]. The WDD makes novel use of optical networks to facilitate a new content-based messaging distributed computing paradigm. WDD is a novel new concept to address the inter-processor communication issues of distributed computing networks often referred to as grids or peer to peer computing. The inter-processor communications of such computing architectures in many cases is severely limited by the throughput performance of TCP, as well as by classic "head of line" blocking problems and "N squared" interconnection issues when many processors try to communicate with each other at the same time. WDD is based on concepts of optical delay storage devices developed in the 1950s, but in this instance applied to large scale dense wave division multiplexed (DWDM) networks. Large nation wide DWDM networks of 100 or more wavelengths have an intrinsic storage capacity of several 10s to 100s of Gigabytes of data, but most significantly, as opposed to traditional optical delay line technologies, they allow hundreds, if not thousands of processors to access the storage device at the same time. With WDD a large scale multi-wavelength network in essence can be considered like a large disk drive, with each wavelength being a separate track and specialized routers with the WDD software acting as record into the DWDM network as a UDP flow. (Figure 1)

When a WDD node receives an incoming packet, the UDP packet TTL is reset and then forwarded to the next WDD node on the network. In this way the originating packets are continuously circulating around the DWDM network. With each UDP packet flow, optional read/write, file name and other possible applications attributes can be record is responsible for maintaining data flow integrity and packet sequence every time the flow circulates around the network. A number of middleware applications are possible to interface the WDD

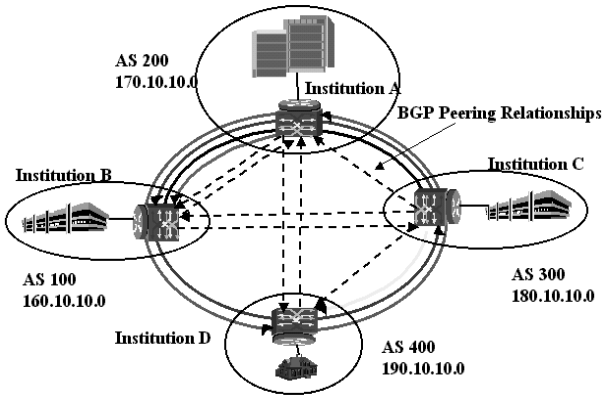


Figure 1 WDD a large scale multi-wavelength network

node with a client computer including a simple TCP emulator, a data workflow control algorithm and ultimately a sophisticated virtual file system. However, a number of distributed computing applications such as SETI@Home, bio-diversity grids, computation fluid dynamic calculations may require only a very simple TCP emulator or workflow architecture since all the records are transient and short lived. Future research activities include using Optical BGP (Border Gate way Protocol) to configure the wavelengths into optical ring configurations for different WDD computing applications and the interconnection of a WDD system into community optical networks which would allow researchers to access the thousands of computing resources large space of n-dimensional cubes whose calculations are to be divided among the processors available in schools and homes public into basic research.

2. WDD ARCHITECTURE

A WDD system would be made up a large scale DWDM system to which are attached numerous WDD nodes. The diameter of the DWDM system has to be fairly large, otherwise the intrinsic storage capacity will be fairly limited. A smaller diameter network, of course, can be compensated by a large number of wavelengths. Instead of the central server waiting for request from a distributed processor to indicate that it is ready to into the DWDM network at the closest WDD node. The datasets continuously circulate and are extracted whenever a distributed processor becomes available. All the central server has to do is to "top up" the flow and always insure that 10 new unprocessed datasets continue to circulate in the DWDM ring. The WDD nodes ultimately could be traditional routers running the WDD software. In the interim the WDD nodes can be simple computers that are connected to

a subtending I/O port on a router that is part of an IP/DWDM network like CA*net 3.Prior to any transmission of data to the distributed computers the WDD node are configured such that each node knows the IP address of the next node in the network. Initially this can be done with assignment of static IP addresses. In a more sophisticated version of a WDD system a management system could assign and keep track of WDD nodes and their addressing. Making Use of the Expected Proliferation of Bandwidth; addressing some of the issues surrounding inter-procesor communication in distributed computing environments, particularly wide-area distributed computing environments.

Making Use of the Expected Proliferation of Optical Bandwidth

The sentiment behind this point is perhaps best expressed following quotation :

- Wavelength division multiplexing (WDM) lambdas [colors] areproliferating quicker than Moore Law, with total bandwidth peroptical fiber increasing faster still-perhaps four times a year.

-Free storage is combining with the explosion of Gigabit and 10 Gigabit Ethernet to make networked storage and caching-storewidth-more effective and less expensive than stuffing your bits under a floppy mattress or in an isolated local hard drive.

-The all-optical network and its increasingly broadband wireless tentacles, are not two or four or ten times as good as the electronic long distance and analog cellular networks they are replacing. Collectively, they are millions of times as powerful.

-The Law of Wasted Bandwidth tells us that the computer that the bandwidth recklessly, will win".

Large Scale DWDM Ssystem

A WDD system would be made up a large scale DWDM system to which are attached numerous WDD nodes. The diameter of the DWDM system has to be fairly large, otherwise the intrinsic storage capacity will be fairly limited. A smaller diameter network, of course, can be compensated by a larger number of wavelengths. The WDD nodes ultimately could be traditional routers running the WDD software. In the interim the WDD nodes can be simple computers that are connected to a subtending I/O port on a router that is part of an IP/DWDM network like CA*net 3. (Figure 2)

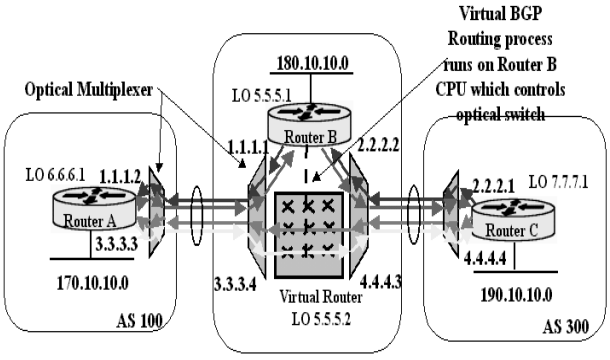


Figure 2 Virtual BGP Routing process

Prior to any transmission of data to the distributed computers the WDD nodes are configured such that each node knows the IP address of the next node in the network. Initially this can be done with assignment of static IP addresses. In a more sophisticated version of a WDD system a management system could assign and keep track of WDD nodes and their addressing.

Injecting Data into the WDD System

One or more client computers are attached to each WDD node. There are two ports/sockets/channels on each WDD node - one for receiving or transmitting data, and the other for control and signalling. The control and signalling channel/port/socket is used by the client computer to signal that it is ready to receive a new set of data from the WDD system, or that it wishes to transmit data into the WDD system. The control and signalling channel also allows the client computer to specify optional attributes of the data set that it wishes to receive or transmit to the WDD system i.e. file name, read/write privileges, etc. When a client computer is ready to send a data set to another distributed computer it signals the WDD on the control/signalling channel where it may also provide the optional attribute information. Once the control and signalling is completed the client computer or server then initiates a TCP session with the WDD node on the data port/socket/channel and transfer the appropriate data that is to be injected into the WDD system. Since the WDD node is in close proximity with the client computer the data and control/signalling TCP sessions can be completed without being limited by the usual TCP performance problems of a long haul network. As well, it is conceivable that a future version of the WDD system these signalling and transfer processes could be part of the TCP stack in the client computer or in the router in the DWDM network, thereby obviating the need for a separate WDD device.

The WDD node converts the TCP packets to UDP packets. Optional attributes that were provided the control/signalling channel are pre-pended to the UDP packet flow and injected into the DWDM network as one entire UDP data flow. The WDD node also maintains a copy of the complete data flow in local memory. The WDD nodes forwards the UDP packets to the IP address of the next WDD node in the network.

Receiving Data at a WDD Node

When a WDD node receives a UDP flow it first checks to see if it was the originator of the UDP flow. If it is the originator itself, it then checks to see if the UDP flow has been marked as "received ok" by some other node. If so, the UDP flow is deleted and removed from the WDD system. If the flow is not marked as "received ok" then the copy of the flow that has been kept in local memory is injected into the WDD system .Re-injecting the original copy will correct for any possible out of sequence packets or corrupted data.

If the WDD node is not the originator of the UDP flow it then checks to see if any subtending client computer has signalled that it is ready to receive the data flow. If not, the UDP packets are immediately forwarded to the next WDD node on the network. The WDD node however does not change the source IP address on each IP packet. As well no attempt is made to check to see if the data has been corrupted or if there are any out of sequence packets. If the receiving WDD node is signalled by a client computer that it is ready to receive a new data set, then the UDP flow is copied into the WDD buffer memory. The UDP flow is checked for data integrity, packet sequence and completeness. If the UDP data flow has found to be corrupted or arrived without of sequence packets then the first packet of the flow is forwarded back to the originating WDD node and the rest of the data is deleted.

Assuming a complete UDP flow has been received, then the WDD node extracts the optional control information provided by the originating computer which has pre-pended to the front of the data set in the UDP flow. Such information could include file name, read/write privileges and other information. If there is optional control information the WDD first sends it

to the client computer through the control signalling channel for acknowledgement to determine if this is the file name or type, rather than the first available data set. If the client computer acknowledges that this is the correct data set, then the data is forwarded to the client computer via a TCP session on the data channel. If it is not the correct data set as required by the client computer, the client computer sends a NACK to the WDD and the UDP flow is re-injected into the data stream and forwarded to the next WDD node.
If the client computer acknowledges that this is the required data set then the WDD node marks header packet of the UDP flow indicating that the UDP flow has been "received ok". The WDD forwards the first packet which contains the "received ok" bit to the originating node. It may elect to forward this packet several times at random intervals to absolutely insure that the originating node receives the acknowledgement, otherwise the originating node will re-inject the original flow back into the network. If there is no optional control information then the WDD immediately marks the UDP flow as ":received ok" and forwards the first packet of the UDP flow back to the originating node

WDD Node Failure and WDD Management
The processing of exception conditions can turn a very simple concept or protocol into a complex and un-scalable solution. For example, the handling of a WDD node failure can cause significant complications if not handled properly. The easiest solution to protect against WDD node failure is to have a set of "keep alive" packets flowing from node to node. With "keep alive" packets and suitable timers, a WDD management system could be immediately notified of any node failure and take corrective actions to maintain integrity of the WDD system. The details of a WWD management system will not be presented here, that has to be addressed to produce a production level WDD system.

3. SOFT RAID OF CLUSTER

One of the main advantages of cluster of workstations is the great amount of resources available in the system (disks, memory, tape units, etc.). Traditionally, these large number of resources were not accessible from all the nodes in the net-work. Only the processes running on the node that had a given resource attached were able to use it. And, if there was a way to access those remote resources, the steps that had to be done were neither simple nor transparent. In a cluster of workstations, these resources have to be accessible from any node in a friendly way. Actually, it would be ideal to be able to share all these resources in a transparent way to the applications. As these cluster of workstations end up being very similar to parallel machines, the same kind of applications can be run on them. The problem we find when executing these applications is that they usually need a high performance I/O system. These applications work with very large data sets, which cannot be kept in memory. They expect a fast filesystem that is able to write and read this data very rapidly.

Using Clusters to Increase the I/O Performance
If a high performance filesystem is to be achieved, we first need to examine the characteristics a cluster of workstations has and the way we should use them to build better filesystems.
The first advantage, and the most obvious one, is the great quantity of resources available in a cluster of workstations. These systems have many disks that can be used in parallel to increase the disk bandwidth. We can also use the large amounts of memory to build big filesystem caches to decouple the performance of the disks from the performance of the filesystem.
The second advantage found in this kind of environments is the high-speed interconnection network that connects all the nodes. Such fast networks allow the system to relay on remote nodes to perform many tasks. We can now use the memory of a remote node to keep the cache blocks that do not fit in the local cache. We can also request other nodes to gather all the I/O requests to build larger ones.

Physical Placement of Data
We will also present solutions to the I/O performance problem from the disk point-of-view. The first thing of which we must be aware is that disks are mainly built of mechanical components that slow down the most common operations (head movement, disk rotation, etc.). Furthermore, the performance of these mechanical parts is not expected to keep pace with the rest of electronic components found in the system. For this reason, the only solution left to increase the disk performance at this level consists of placing the data in such a way that the mechanical parts have as little effect as possible on the global disk performance. In this section, we will address different ways of placing data on the disk (or disks) to increase the performance of the I/O subsystem. Some of the solutions presented here are especially targeted to parallel/distributed systems, while others were already used in traditional mono-processor systems but are frequently used in I/O subsystems for clusters of workstations.

Software RAID
Clusters of workstations may have one or even several disks attached to each node. This property can be used to increase the bandwidth of the I/O system. The idea is to distribute the data among the disks so that it can be fetched from as many disks as possible in parallel. This will increase the data transfer bandwidth as many times as disks are used in parallel. The idea was to connect several disks to a single controller and give the impression that the disk had a higher data transfer bandwidth. This kind of disks is currently known as RAIDs (Redundant Arrays of Inexpensive Disks).

High Performance of a RAID
The high performance of a RAID is mainly due to three reasons. The first one is that data from each disk can be fetched at the same time, increasing the disk bandwidth. The second one is that all disks can perform the seek operation in parallel, decreasing its time, which is one of the most time-consuming disk operations. Finally, in some kinds of RAIDs, more than one request may be handled in parallel, which also increases the whole system performance.
As we want to be able to access as many disks in parallel as possible, the data must be distributed over the disks adequately. Interleaving the data among the disks seems to be the right way to do it. Using this distribution, if the request is big enough, each disk will keep at least one block of the request, and data from all disks will be fetched in parallel, achieving the highest possible bandwidth. This data interleaving can be characterized as either fine-grained or coarse~grained. Fine-grained disk arrays conceptually interleave data in relatively small units so that all I/0 requests, regardless of their size, access all the disks in the disk array. This results in very high data transfer rates for all I/O requests. On the other hand, it has the disadvantages that only one logical I/O request can be served at any given time and that all disks must waste time positioning for every request.

Coarse~grained disk arrays interleave data in relatively large units so that small I/O requests need only to access a small number of disks and large requests can access all disks. This allows multiple small requests to be serviced in parallel while still a lowing large requests to achieve high transfer rates. Furthermore, if many small requests are served in parallel, all seek operations are also done in parallel, while on a fine-grained RAID they must be done consecutively.

Another important design issue is to achieve some degree of fault tolerance. As many disks are used, the probability of a failure in one of the disks is quite high. This means that a RAID needs a fault-tolerance mechanism to allow a disk failure without losing the information kept in the failed disk. This tolerance is achieved introducing redundancy. The way this redundancy is implemented along with the striping granularity is what distinguishes the five levels of RAIDs

A RAID level 1 uses half of the disks in the array to keep a copy of the other half. Whenever data is written on a disk, it is also written to its redundant disk. When retrieving the data, the disks which need the smallest seek are used. The problem with this approach is that half of the disk space cannot be used to store useful data. There are RAID 1~5. In order to shortage this paper , we only introduce RAID level 5. RAID level 5 is like the previous one but eliminates the parity-disk bottleneck by distributing the parity uniformly over all of the disks. With this modification, all disks can be used in read operations, increasing their performance. Write operations can also be done in parallel as the parity blocks are distributed among all disks. This version has the best small read, large read, and large write performance of any redundant disk array. On the other hand, small writes are somewhat inefficient compared with other redundancy schemes.

4. CONCLUSION

The possibility of a WDD system for distributed computing applications opens a number of other interesting new concepts in terms of e-research, optical ring management and ultimately the democratization of basic research. The implications of this concept on telecommunications manufacturers and carders could be quite significant. This novel use of optical network technology could also become an important use for the predicted glut of bandwidth that overhangs the market today. Although methods of Data Storage Using WWD & Cluster Software RAID are different , we can get more information from this paper. No longer would one think of networks as simply a means of computers exchanging data directly with each another, but in essence, ***as the saying goes the network is the computer.*** The first can be characterized as an opportunity, a current resource which is expected to grow significantly in the coming years. Whereas second can be characterized as a requirement, real-world limitations technology application needing solutions.

5. REFERECES

[1] Bill St .Arnaud, CANARIE Inc, Wavelength Disk Drives http://www.ccc.on.ca/wdd/wdd_home.htm.

[2] Anurag Acharya and Sanjeev Setia Using Idle Memory for Data-Intensive Computations. In *Proceedings of the 1998 ACM SIGMETRICS Conference on Measurement and Modeling of Computer Systems*, June 1998.pages 278-279.

[3] E. Anderson and J. Neefe. An Exploration of Network RAM. *Technical Report CSD-98-1000*, Computer Science Division, University of California, Berkeley, July 1998

[4] Kai Hwang ,Scalable Parallel Computing :Technology, Architecture ,Programming ,WCB McGraw-hill ,1988

[5] T. E. Anderson, D. E. Culler, and D. A. Patterson. A Case for NOW (Networks of Workstations). *IEEE Micro*, vo1.15(1), February 1995 pages 54-64.

The Distributed Object-Oriented Technology In Undergraduate Education of Computer Science

Diao Cheng_Jia
Information and Technology College, Nankai University
Tianjin 300071, China
E-mail: Diaocj@office.nankai.edu.cn

ABSTRACT

Based on UML, the technology of the object-oriented analysis (OOA) and the technology of object-oriented design(OOD) have been characterized by the generalization, visualization, the rapid development of the software and the easy maintenance, so that the object-oriented technology has become the popular methods in developing the large software project. However, the undergraduate education of computer science in the domestic universities only covers the studies of the object-oriented programming languages (Such as C++ or Java). In the course of software engineering, the object-oriented methods are taught as a part of the course and introduced shallowly. In the phase of graduate education, the system training is conducted. However, the distributed object-oriented technology is not taught yet. Obviously, there is a contradiction between the current situation mentioned above and the requirement to the large number of the qualified project analysts and designers, and those, who have grasped the OOA and OOD, are wanted urgently in the market. In this paper, the OOA & OOD training are discussed and some suggestions & proposals about the necessity and feasibility are also put forward.

Keywords: distributed, parallel processing, the object-oriented technology, education on computer

1. INTRODUCTION

Information can be regarded as the important strategic resources for the development of society. The traditional ways of producing, managing and living are changed by the information technology and information industry, which make the economy growing rapidly. Information network can be characterized by internationalization, socialization, openness and personalization, which make the information frontier of every country extending to every unit and every person connected to Internet. We can say that it almost has reached every corner of the world. With the rapid development of information technology, many kinds of information system are adopted in telecommunication, electronic business, commercial network etc, so that our economy can be integrated with the world economy.

Internet (world wide network) in our country born from network for research and education, has become the commercial network, which means that we have entered the new era of Internet. The quantity of users connected to Internet has been increasing in a large scale yearly, and now it has reached tens of million. Furthermore, it is increasing day by day. The architecture of LANs has developed from the hierarchical structure to the distributed structure. The relative algorithms and the new software come one after another. The invention of the software components will make the collaboration among system more important.

With the rapid development of Internet, network security becomes more and more important. Nowadays, the network resources are usually organized according to the scattered, distributed architecture. Internet is made up of many LANs, which are connected by gateways. The resources in a LAN are shared, and the gateway becomes the focus of security in a LAN. Computers inside a LAN have the equal level of security, which means that if the gateway is broken, the whole LAN will be broken. Therefore, the distributed parallel architecture should be adopted in intrusion detection system. Every server and computer should be monitored by the relative detection software. The scattered detection system should not only complete their task respectively, but also cooperate with each other so that the intrusion detection can be accomplished. At present, the Agent technology has become the direction in which the distributed, parallel intrusion detection system for the large scale network has been researched. It will have a bright future. The system can be completed by means of OOA & OOD.

The OOA & OOD based on UML, characterized by generalization, visualization, standardization, development team, free distribution & combination, as well as security, rapid development and easy maintenance of the software product, has become the popular method for software development. A great deal of software engineers are wanted, and especially those are needed urgently, who are qualified to be project analyst and designer in the background of today's IT development. Actually, project managers, who have grasped the OOA&OOD, are wanted in the current market.

It is well known that most of software development talents graduate from computer department of a university. Therefore, every computer department of every university should consider how to produce the software development talents wanted in the market as soon as possible. In this paper, some suggestions and proposals on how to grasp OOA&OOD will be put forward.

2. OBJECT-ORIENTED TECHNOLOGY

The Object-oriented programming can be regarded as a milestone after the time of structured programming in the development of software technology. Although the object-oriented programming is based on the programming languages, its concept can not be expressed by any programming language. The large and complicated software system can be divided into many object components by means of the object-oriented technology. In addition, the objective facts can be simulated clearly and briefly by using object models. The object models can be reused easily. The relative components in a system can be depicted clearly in the object-oriented architecture.

Software is used widely such that software industry has been developing rapidly. The object-oriented methodology can be regarded as the best option to meet up the increasing requirement to software, to produce software in a short period of time, and guarantee quality, reliability and easy

maintenance. With the development of software, new algorithms, data structure, architecture and new technology come into existence one after another, such as, the distributed computing technology in network, prototypes, the popular components, and component technologies etc. All of these new technologies can be included in the object-oriented methodology. As time goes on, these new technologies and methods should become a part of the object-oriented technology. Software tools can be updated continually and programming languages have been developing. However, the core of the object-oriented design and its architecture will not become obsolete. If you have grasped the arts of the object-oriented design, the programming design will become the pleasant task. Furthermore, software engineers can also benefit from OOD.

Since 1980s, the object-oriented methodology has been the trend of software development. During the 5 years from 1986, the number of the object-oriented methodology and the relative modeling technologies increased from 5 to more than 50. At the end of 1997, UML(Unified Modeling Language) came into being, which can be regarded as the integrity of many modeling languages, and also can be used to describe standardization and visualization of software. Here, Unified means integration. In UML, the following items are unified:

1) The various modeling languages are unified;
2) The development phases are unified;
3) The different application areas of software are unified;
3) Internal structures are unified;
5) New technologies are included, such as the distributed computing in network, prototype, the popular components and components technology etc.
6) It can coexist with many different development procedures.

Generally speaking, different software companies use different ways to develop their software, but all of them have accepted UML as the standard for software development. At present, the so-called object-oriented technology actually refers to the object technology in which UML has been adopted as the standard.

Up to now, only several large systems have been completed by means of UML in China. Therefore, a large number of qualified analysts and designers are wanted in our software industry, especially those who have grasped OOA&OOD. Therefore, more consideration should be taken on how to strengthen system training of the object-oriented technology in our undergraduate education of computer science, and produce a great deal of talents needed in market as soon as possible.

3. THE OBJECT-ORIENTED ECHNOLOGIES IN THE EDUCATION OF COMPUTER SCIENCE

According to the object-oriented technology, the procedure for developing a software system can be described as follows:

1) OOA (Object-Oriented Analysis)
2) OOD (Object-Oriented Design)
3) OOI (Object-Oriented Implementation);
4) OOP (Object-Oriented Programming):only part of OOI;
5) OOT (Object-Oriented Test);
6) OOM (Object-Oriented Maintenance).

From the introduction above, we can see that the Object-Oriented technology is a whole process. It means that you can not complete an Object-Oriented software system if you only grasp OOP. In our computer education, the Object-Oriented programming language(C++) is taught in the first year and another Object-Oriented programming language (Java) is taught in the third year or fourth year. In the course of software engineering, the Object-Oriented method is taught only as a part of course, and introduced superficially. The systematic training is conducted only in graduate education.

There are many applications of network architecture designed according to the distributed object-oriented schema, and there are also many software components on the relative system monitoring. It has been the focus of software industry recently in the world. The undergraduates have less access to the distributed object-oriented technology. Thus, the undergraduates are trained only to be qualified as the junior programmers or the advanced programmers, but not to be qualified as the object-oriented system analysts and designers. However, a great deal of qualified project analysts and designers are wanted in domestic computer software industry and furthermore, project managers, who have grasped OOA & OOD, are needed greatly.

Although there is a bright future for the object-oriented technology, the latent power of it cannot be realized easily if some techniques are not used correctly. In order to make the object-oriented technology more useful by the training of OOA & OOD, and improve students' ability to use the technology, this course is taught. More and more people begin to realize that these techniques are the critical factors in development and maintenance of the important projects.

It is necessary and feasible that the training of OOA (Object-Oriented Analysis) and OOD (Object-Oriented Design) are conducted in the undergraduates' education of computer science. This course can be taught with the course of software engineering in a same term and can be completed in 40 hours. Students can learn the object-oriented methods for better OOD by taking this course in which a set of explainable design rules and the heuristic methodology will be adopted. By studying and using these information and technologies, students can learn to understand problem more skillfully from concepts and procedures, and can work out a better solution by means of the object-oriented technology.

Based on UML, OOA & OOD can be viewed as a standard in modeling technology, which can be independent of any special programming language. Certainly, as soon as the model is built up, it will be very easy for you to convert symbols in the model into any programming language (such as C++, Java, Visual Basic, IDL etc) with which you're familiar. The conversion can be done by hand, or by using the familiar software tools, ie, CASE(such as ROSE by Rational). It will not very difficult for the senior students to grasp OOA & OOD, since they have already mastered an object-oriented programming language.

In the course of the object-oriented technology, the items below will be covered:

1) UML: Unified Modeling Language.
2) General prototype for static design of objects
3) Persistent objects and object-oriented design and component technology.
4) General prototype for interface design
5) The object-oriented architecture

In the course of teaching and learning, a large number of uses and prototypes for the object-oriented design can be used as exercises so that students can understand and master what they have learned as soon as possible. If the condition is OK, the student can have chances to put their design into practices such that they can grasp the knowledge more deeply. The scores will be given according to the result of the closed examination or the perfection of their design.

4. CONCLUSION

Up to now, the object-oriented technology has been developing for 16 years. Especially, in 1997, UML came into being, which can be regarded as the token for the maturity of the object-oriented technology. There are many large software projects completed by means of the object-oriented technology. Therefore, there are plenty of materials books and teaching materials relative to it. In China, only several large scale software systems are completed according to UML by means of the object-oriented technology.

Recently, more and more books about the object-oriented technology come into being. The relative material and books also come into existence. Most of books about the object-oriented technology belong to the translated version. It is not only feasible, but also very necessary to teach the object-oriented technology to the senior students majoring in computer, since computer talents are wanted greatly and the relative material are available everywhere.

5. REFERENCES

[1] Yang zhenfu The object-oriented Analysis and the Design. China Railway Publishing House 2001.4

[2] RunDong Liu Object Design & Programming with UML 2001.2

[3] Grady Booch James Rumbaugh Ivar Jacobson The Unified Modeling Language User Guide 1999

[4] zhang long System Analysis & Design with UML .People's Post and Telecommunication Press 2001.8

[5] Mark Priestley Practical Object-Oriented Design with UML , The McGraw-Hill Companies , Inc. 2000.3

Modern long distance teaching and network curriculum architecture

Tan Ran Xue Shengjun Yin Fan Kang Ruihua
Institute Computer science and Technology, Wuhan University of Technology
Wuhan, Hubei, 430063, China
E-mail: tanran@public.wh.hb.cn

ABSTRACT

The paper introduces the architecture of modern long distance teaching resource system and design requirements of the network curriculum. And the environment of resource application and management will be presented in the paper.

Keywords: modern long distance teaching; long distance teaching resource; network curriculum; resource management

1. INTRODUCTION

Applying modern long distance teaching engineering to form open education network and constructing a lifetime learning system is an important engineering to utilize and optimize education resource, to popularize and enhance basic quality of all the people, to reduce education costs, and to make everyone to enjoy the right of education. With the education software suit for remote transmission and interactive study as textbooks, modern long distance teaching engineering is a new management style of the modernization education industry by means of modernization information technology.

The core of building modern long distance teaching engineering is the education resource building. The education resource building has four layers of implications: 1) the education resource building of source material, including the problem base, the source material base, the courseware base and the case base; 2) the network curriculum base building; 3) the education resource management system developing; 4) the support platform of general long distance teaching system developing. In these four layers, the network curriculum building and the source material education resource building are of the emphasis and core, meanwhile the third and forth layers are the facilities layer building. Specific contents of the network curriculum and the source material vary from each other and have each characteristic. Correspondingly, the management system and education system must be suit for the variance forms to make most of their characteristics.

2. MODERN LONG DISTANCE TEACHING RESOURCE SYSTEM ARCHITECTURE

Being not the digitment and networkment of the traditional education, modern long distance teaching depends on the education support platform to carry out remote teaching and learning. Its final aim is to support many kinds of teaching styles, especially the exploring and collaborating teaching style with students as the center, so as to cultivate creatable talents. Therefore, to undertake the activities of teaching, learning, doing homework, answering questions and testing, we must apply the special network education support platform which is important to provide the convenient, flexible teaching design and organization in accordance with long distance teaching characteristic; and we should apply and manage teaching style, schooling mode, learning strategy, teaching evaluation and tracement efficiently in accordance with the characteristic of each teaching link.

Modern long distance teaching resource constructing includes the source material base, the problem base, the case base, the courseware base, the network curriculum building, as well as the development of the teaching support system and the modern long distance teaching management system, which are suit for many teaching styles. The contents and the relations among them constitute the modern long distance teaching resource architecture. In this architecture, CERNET, satellite TV education network and Internet network support environment is the platform of modern education resource building, applying, and running. Of all the resource, the media source material base is the base. Coursewares in the courseware base, cases in the case base, the network curriculum and even the problem base may use the media data in the media source material base. Several knowledge points' courseware or different teaching link courseware and the self-test or test problem base synthesizes the network curriculum. Each resource base aforesaid has built index information, for the use of quick selection, browse and access.

In network teaching all the communication and intercourse and the critical teaching links need be supported by specialized tools, which, however, the Internet technologies have not provided, so we should develop network-teaching tools. In addition, it is difficult for the non-computer major teacher to design the network interactive programs, hence, we need develop a set of network teaching support platform to provide overall teaching support tools for teachers to carry out network teaching, to mask programming complex, so that teachers may centralize in teaching, and the network teaching may change from the simple teaching information distribution to an interactive subjunctive study community. The network teaching support environment specially means software and teaching resource that support network teaching, and the teaching activity carried out on the network-teaching platform.

The teaching support system comprises a series of teaching tools that support many teaching styles, including the learning system (real time/non-real time), the teaching system (non-real time/real time), the teaching resource edition system, the coaching and answering question system, the viewing homework system, the test system, the evaluating system, the intercourse and discuss tools, the subjunctive experiment system and the searching engine etc. These teaching and learning tools, based on the long distance teaching resource base, are used to accomplish each teaching and learning activity of long distance teaching, in order to achieve remote collaborate. The learning system, the teaching system, the education resource edition and the fabrication system may relate to

the media source material base, the problem base, the courseware base, the case base and the network curriculum possibly, and the test system relates to the problem base, while the evaluation system involves every part of the education resource.
To carry out modern long distance teaching favorably and effectively, efficient management is important. The modern long distance teaching management system comprises the resource base management (media source material base management, the problem base management, the case base management, the courseware management, the network curriculum management, etc.), the education management (the teachers management, the students management, the students' record management, the education affair management, the learning management, the test management, etc.) and the system management (the security management, the performance management, the costing management, the malfunction management, etc.).

3. THE DEMAND OF MODERN LONG DISTANCE TEACHING ON NETWORK CURRICULUM

Modern long distance teaching has broken down the traditional school education style of teaching in class; moreover it has broken through the limitation that the traditional education can't carry out efficient communication and intercourse. It needs to build a new teaching and learning style. The style does not only arrange teaching contents in network for students to view, but also demands the full communication and intercourse between teachers and students via network, which makes students feel to study with the direction of teachers, but not to learn from computers, that's to say, there be humane intercourse.
The second demand of modern long distance teaching is that teachers organize and help students' study via network. Teachers should do their best to make students take part in study on their own, and pull students out of difficulty timely during the course of study, so that the core links of traditional teaching, such as teaching, learning, testing, doing homework, discussing, evaluating, answering question library, noting, can be fulfilled via network.
The third demand of modern long distance teaching is that the network should provide the satisfactory teaching evaluation and the diagnosis tools, so that teachers can know about learners' main feature and study development, provide individual help, and improve teaching.
The forth demand of modern long distance teaching is to carry out the basic quality education via network, convert traditional knowledge education to ability education, and require network to support many kinds of education styles, especially the problem-based exploring and collaborating study style; provide traditional campus culture atmosphere, humanity spirit, extend campus potential effect in space, and provide inner spirit materialized tools, such as the library, the subjunctive corporation, the subjunctive academic lecture , students community, teacher community etc.

4. MODERN LONG DISTANCE TEACHING MANAGEMENT SYSTEM

(1) Rules of System Design
The design of modern remote teaching management system cares about the expandence and performance design of hardware system, and security, dependability of software system.
The system runs in Internet environment, and the software, hardware platform is move and more various. To meet the need of different system, the development of management software must keep with the rules of open system, not the platform. Follow common international standard, so as to be easy to upgrade maintenance.
To use large model business database system is good to rise throughput time of abundance data, and it is good to standardization of manage about system, and it can to ensure data integrality and security.
The software system structure is browser/ server account pattern, the system server end is expansible CORBA/EJB (distributed servers account pattern),user disparted module arrangement structure , many module schism , allow distributed system to process parallelly , and so the efficiency is better.
In hardware environment and system software, users running environment support Chinese Netscape4.0 or IE4.0. Server system software support distributed computing UNIX server system or Windows NT system.
Management system requires universal software system used in the same system platform. There are install program, users directions and technical report.
Material base management system is flexible in data management. It can manage all the material in long-distance teaching project, also can class the material in form of style or subject, also can cut large material base to small material base and then sale them.

(2) Management of Teaching Material Base
The basic management of teaching resource base includes the management of media material base, problem base, index base, courseware base, network course base, cases base, and the retrieve. The education contents management should ensure the problem base, courseware base, network courses base, and cases base security, dependability, secrecy, it can provide download and compress download function for the good teaching contents, support the biggest parallel accessing function, ensure the system extensible. The teaching contents transmission function support multimedia upload and download mainly, ensure the multimedia transmission security, stability and secrecy, integrate all kinds of nowadays ripe technologies and products, ensure the transmission reliable and timely.

(3) Remote teaching system management
The remote teaching system management includes education evaluated system, user management system, and network management system. The remote teaching evaluated system consist of teaching resource base evaluated system, teaching supported software evaluated system, charge evaluated system; the user management system includes the function of user group management, user management on the basis of grouping, user register and user accounts management, user authorization and authentication management, maneuver information management, audit management; network management system provide function of network malfunction management, network configure management, network performance management, network charge count management and network safety management .

5. THE APPLIED ENVIRONMENT OF NOWADAYS LONG DISTANCE TEACHING

Nowadays remote teaching applied environment as follows chart, use Internet, CERNET coupling all college campus network, realize teaching resource high share, all network courseware run one of three software platform as Fig.1.

● run the network courseware on stand-alone, it must run on Windows 95 or higher edition.

● some courseware which on the basis of static web or alternation courseware which on the basis of server release must be capable of access using the standard WEB browser, but is independent of the hardware platform which the browser run on.

● the dynamic courseware that was exploited on Java, it must be capable of running on the standard Java virtual machine, but is independent of the hardware platform of the virtual machine.

The applied environment of teacher involved mainly prepare for lessons, prelection, linking PC, exploited tool, teaching material base, courseware base, the classroom of video conference system, network courses, electronic teaching plan, the communication with students, teaching supported environment, E-mail, BBS etc.

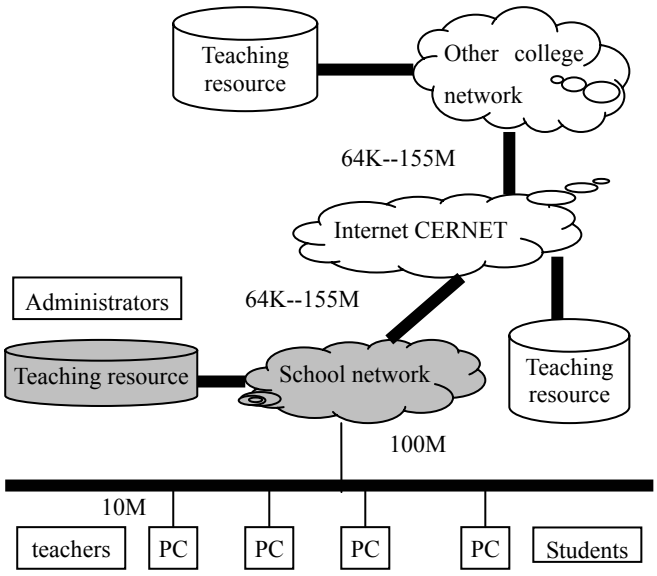


Fig.1 Software platform of network courseware

Students apply environment includes listening, listening system video conference system, study by oneself, linking PC, accessing teaching network online, network course, teachers communication with the other students, E-mail, interactive electronic classroom, teaching supported environment, BBS, examination, online examination, problem base, etc. Application environment for educational administration managers should provide education material management system, student record management system, class-select system, teaching supported environment, problem base, teaching quality monitor, etc.

6. REFERENCES

[1] Xue Shengjun Tan Ran Huang Hua. CSCL: An Internet Based Education Model CSCWID'2001.

[2] Xue Shengjun,Gao Xiaohong,Tan Ran. Multi-Agent System Architecture Research. CSCWID'98.

[3] Xue Shengjun,Gao Xiaohong,Jiang Jingjue. The Mechanism of Multimedia Communication Synchronization in the CSCW Environment. CSCW'97.

[4] Tan Ran Xue Shengjun. An Autonomous Agent and Recognition Based on Self-organizing Neural Architecture. Proceedings of International Workshop on CSCW in Design

[5] Lin Zongkai,Jean-Paul Barthes. Proceedings of International Workshop on CSCW in Design.

[6] David A.Robinson and Callum R.Lester and Neil M.Hamilton, Computer Networks and ISDN Systems 30(1998)301-307

[7] Cornell.G and Horstmann,C.S, Core Java.SunSoft Press(Prentice-Hall), California, 1.1996, pp.12-16,177

[8] Soloway E, Technology in education. Special Issue Comman ACM 36.No.5(1999)

[9] Henry MeLoughlin, WEST,Computer Networks and ISDN Systems 28(1999)1887-1890

Research and Design of an Off-line Portable Scanner Based on DSP

Junning Chen
Computer Science and Imformation Technology College,
Anhui University, Hefei, Anhui, China, 230039
E-mail: jnchen@mars.ahu.edu.cn
Yuehua Dai
Computer Science and Imformation Technology College
Anhui University, Hefei, Anhui, China, 230039
E-mail: daiyuehua1975@hotmail.com
And
Daoming Ke
Computer Science and Imformation Technology College,
Anhui University, Hefei, Anhui, China, 230039
E-mail: ke-daoming@163.net

ABSTRACT

As a computer's graphic and image input equipment, scanner has been applied to the fields such as graphic and image process, publishing and printing, office automation, etc. But now on the market, a common character of all kinds of scanners is that the application is based on line to computer, so on some occasions this is not convenient for the users.
This paper presents a design method of an off-line portable scanner, which can be used without computer. We discuss the system based on a synchronous Digital Signal Processor with multicenter. TI's TMS320VC5402 is a sixteen-bit fix processor[1], and we make it the core of our control system. Because it can not only handle data with high speed but also manipulate the data by abundant arithmetic/logic operations and determine the results of process which can be output as images very quickly. This controller can implement different scanning controls for various common portable scanners of 100 to 400 dpi (dots / inch)[2]. It can communicate with PC through LPT, and fulfill the further display, analysis and handling of the scanning image data. The off-line portable scanner is easy to carry and can deal with the image on-the-spot.

Keywords: Off-line, Portable, Scanner, DSP, HPI

1. INTRODUCTION

With the growth of technology and the increasing improvement of people's living standard, computer is more and more common in family. However, scanners also become common in family because the price of scanner has decreased a lot for the past years. At the same time, digitalization becomes global. As one of the most important basic technology, digital signal processors are applied in many aspects of most fields, and their applications are increasing with un-preceding speed. Because digital signal processors are one of the most important devices that can process signals in real time and with high speed, they have received increasing attentions from the fields of science and engineering. And as we know, the technology of digital signal processors is quite difficult new high-technology both from the views of theory and of the engineering application[3,4]. According to the market needs and the direction of technology development, we apply the technology of digital signal processors on scanners.
This paper presents a design method of an off-line portabl e scanner, which is based on a synchronous digital signal processor with multicente and can be used without computer. We discuss how the hardwares design and give the software process. The scanner system consists of three parts, and the whole structure of the system is shown as the Figure 1.

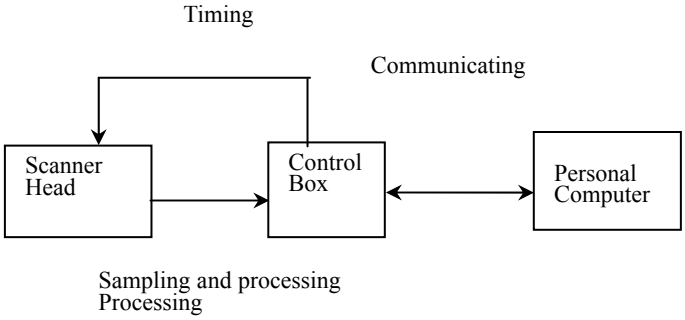


Figure 1 The Structure of Scanner System

2. THE MODEL OF SCANNER HEAD

Through a cable in which there are eight threads, scanner head output its signals to a special interface which is tied to and is a part of the Scanner Head[1,5]. The structure of the interface is shown as Figure 2.As we see in Figure 2, there are eight colored threads, and the functions of each thread is described in the legend. The scanner head can scan the biggest width of 105 millimeter, and what it scans can be a character style and three kinds of image styles. These characters and images have four kinds of quality such as100 dpi, 200 dpi, 300 dpi and 400 dpi..
The four kinds of quality also can be respectively expressed as 400 image unit per scanning line, 800 image unit per scanning line, 1200 image unit per scanning line and 1600 image unit per scanning line. A voltage of DC 12- volt is applied on the scanner head, and the largest current we measured is 400 milli-ampere[6,7].

3. THE MODEL OF CONTROL BOX

In our control box, there is a digital signal processor, '5402 (TMS320VC5402), an external memory-Am29F400B, a reset circuit , a timing circuit and so forth. Especially, the digital signal processor, '5402 is a key device in the control box. According to our design, the control box need fulfill three functions: First, to receive, process and store the image data

scanned by the scanner head. Secondly, to shut the power applied on the CCD (Charge Coupled Device) every 30 seconds in order to protect the photoelectric element in thescanner head. Finally, to communicate with PC (personal computer) so that the PC can carry out the further display, analysis and handling of the scanning image data. In fact, the three functions of the control box are fulfilled by '5402. We adopt DMA (Direct Memory Access) based on McBSP (multichannel buffer series ports) of '5402 to realize the sampling and transferring of the image data. The receive channels of the McBSP can transform series signals to 16-bit parallel signals. That is a very important character of '5402, and other processors, such as singlechip, must turn to a special external circuit to realize the transformation. On this point, the McBSP of '5402 simplify the hardware design. The DMA embedded in the '5402 can make the data transfer more convenient[6,7]. Because, after assigning the controller of DMA, the series data can be transferred continuously regardless of CPU. So, we can greatly improve the system efficiency by adopting the DMA.

The process of tansform from series signals to parallel signals is described as the follow:

Data arriving on DR is continuously shifted into RSR. Once a complete word is shifted into RSR, an RSR-to-RBR transfer can occur only if an RBR-to-DRR copy is completed. Therefore, if DRR has not been read by the DMA since the last RBR-to-DRR transfer (RRDY=1), an RBR-to-DRR copy will not take place until RRDY=0[8,9,10]. At this time, new data arriving on the DR pin is shifted into RSR and previous contents of RSR is lost.

Worth of saying, CPU does not affect DMA transferring image data. Obviously, there are many benefits, and one of them is to decrease greatly the need of external memory. When the data in a buffer reach a specific number, DMA will inform CPU of handling the image data as well as continuously transfer the image data to the buffer, and then CPU will transfer the handled data to the external memory. As we see, it is more effective than CPU receives image data and then handles them.

The '5402 have two on-chip timers. The on-chip timer is a software-programmable timer that consists of three registers (TCR, TIM and PRD). The high dynamic range of the timer is achieved with a 16-bit counter with a 4-bit prescaler. The timer operates as follow:

The timer is an on-chip down-counter that can be used to periodically generate CPU interrupts. The timer is driven by a prescaler that is decremented by 1 at every CPU clock cycle. Each time the counter decrements to 0, a timer interrupt (TINT) is generated and the down-counter is reloaded with the period value[11].

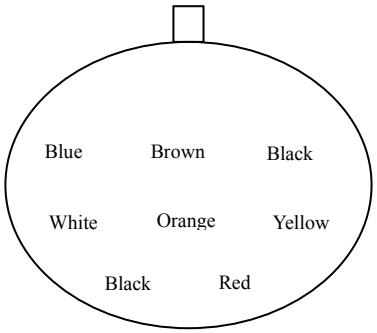
We use timer0 as our timer, and use timer1 as our counter. Once the control system works on for 30 seconds, the timer0 can produce an interrupt signal on the TINT0 pin, and then TINT0 drives CPU to send a signal out to the TOUT0 pin. We use TOUT0 to control the external timing circuit so that we can stop supplying power on the scanner head.

The timer that timer0 can control is limited when the resolution of CPU is 10 ns. The timer0 interrupt (TINT0) rate is equal to the CPU clock frequency divided by two independent factors[12,13]:

$$TINT0_{rate} = \frac{1}{t_c(c)\times\mu\times\nu} = \frac{1}{t_c(c)\times(TDDR+1)(PRD+1)} \quad (1)$$

.

In the equation, $t_c(c)$ is the period of CPU clock, μ is the sum of the TDDR contents plus 1, and ν is the sum of the PRD contents plus 1. PRD is a 16-bit memory-mapped period register, and TDDR is a 4-bit timer divide-down register. So we can get the largest value that the timer0 can control[14].



1. Black : Ground
2. Red : positive pole with 12 volt
3. White : synchronous control signal
4. Orange: control signal
5. Yellow: control signal
6. Blue : CCD image signal
7. Brown : synchronous line signal
8. Black : Ground

Figure 2 The interface of Scanner Head

$$t_{\max} = t_c(c)\times\mu\times\nu = t_c(c)\times(TDDR+1)\times(PRD+1) = 10^{-8}\times(2^4+1)\times(2^{16}+1) \approx 1.1\times10^4 \quad (2)$$

Because the difference between $t_{\max}$ and 30 seconds is too large, we have to use loop programs to time for 30 seconds. We can see that timing for 30 seconds is not very exact. So we need not think about the time that the loop programs take up. Because of that, we load TDDR with 16, and load PRD with 62500. Then, in each circle, the timer0 can only time for TOUT0 can fail the power of the timing circuit.

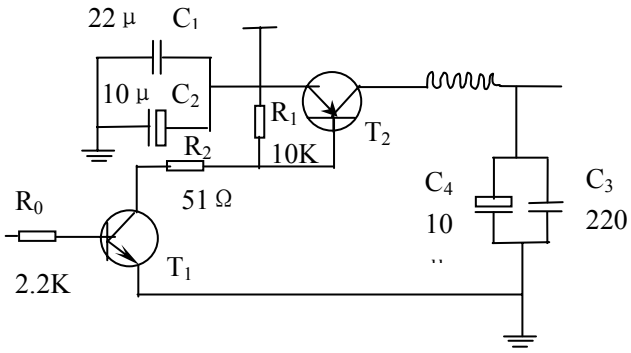


Figure 3 Timing Circuit

We need to point out that the signal on TOUT0 is high voltage and it must be reverted before it is tied to the external timing circuit[15]. The external timing circuit we design is shown as Figure 3.
As we can see, only the low voltage can stop supplying 12 volts on the scanner head.

4. THE MODEL OF COMMUNICATION WITH PC

In the control system ’5402 communicate with PC through a HPI-8. The HPI-8 is an 8-bit parallel port that interfaces PC to the ’5402. Information is exchanged between the ’5402 bidirection data bus and various control signals. Sixteen bit transfers are accomplished in two parts with the HBIL input designating high or low byte. A PC communicates with the HPI-8 through dedicated address and data registers, which the ’5402 can’t directly access[16]. The HPI control register, which is accessible by both the PC and the ’5402, includes bits for configuring the protocol and for controlling communication (handshaking). A simple block diagram of the HPI-8 is shown in Figure 4[17].
The host port interface (HPI) unit is a peripheral to the ’5402, and it can communicates with PC independently of the ’5402 ,or vice versa when required. The interfaces contain minimal external logic[18], so our control system is designed without increasing the hardware on the board. The HPI interfaces to PC parallel ports directly with simple and minimal hardware.

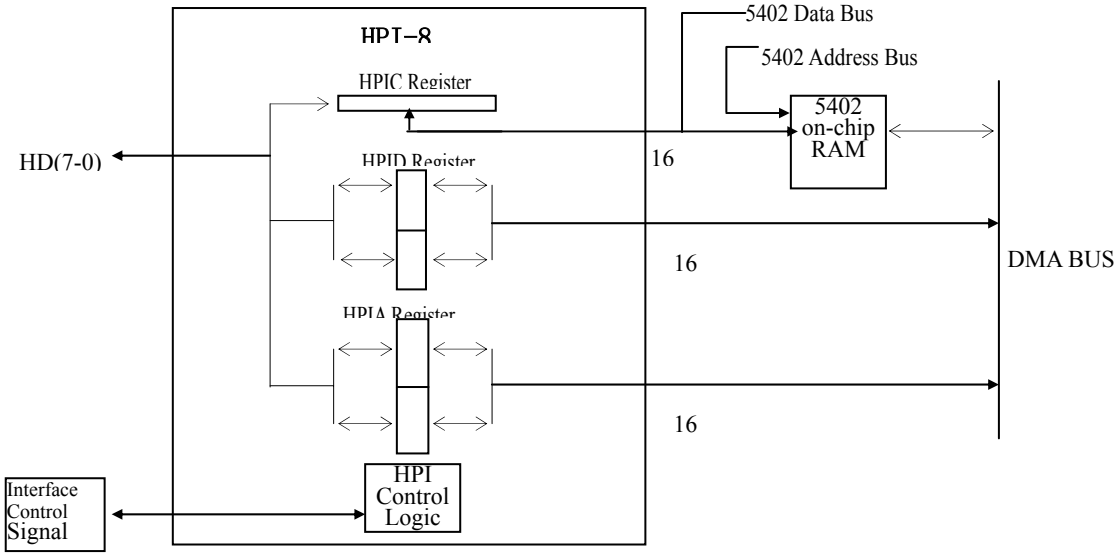


Figure 4 A Simple Block Diagram of HPI-8

Figure 5 shows the HPI design of communication between scanner and PC through HPI. PC accesses data by an R/W control pin, HRW, and a data strobe, HDS1. HDS2 is tied to Vdd. The strobe, HDS1, inputs the data at the rising edge of the signal. The PC parallel port is the host with a separate data and address bus[19]. HAS is tied to Vdd. The falling edge of HDS1 strobes the control signals, HBIL, HCNTL0, HCNTL1, and HRW into the HPI. To permanentlyenable the HPI, HCS is tied to ground. Due to an insufficient number of outputs on the parallel port, SELECTIN and AUTOFD combine with external hardware to create four signals: HCNTL0, HCNTL1, RS(reset), and HBIL[19]. These four signals are input to the ’5402, and the ’5402is reset by generating clock edges from AUTOFD[20]. Data is strobed into the HPI data/address register on the rising edge of HDS1. The HBIL input indicates whether the byte received is the first (two) or second (high) byte. HBIL transitions during the rising edge of HDS1, at a point later in time than HDS1[21]. This delay is the propagation delay of the flip-flop (see Figure 6).The flip-flop ensure that the correct value of HBIL is sampled for every byte transfered.
To ensure HBIL is initialized high at power up, a resistor and capacitor are connected to the flip-flop[22]. The software monitors the HBIL input to ensure the validity of a data/address transfer.
The timing diagram of Figure 6 shows the control signals and data strobed in and out of the HPI. HAD, HD read and HD write are the read and write data signals, and HAD stands for HCNTL0, HCNTL1, HBIL and HRW. The signals show the data transferred across the HPI data bus (HD0----HD7). At time (a), the control signals, HCNTL1, HCNTL0, HRW, and HBIL are set to the required logic level. This level indicates the type of access necessary, whether read or write, and the appropriate registers. At time (b), the falling edge of HDS1 causes the signals to latch into the HPI and set the HPI to the required mode[23]. At time (c), the rising edge of HDS1 indicates data is being written to the HPI or read from the HPI.

5. SOFTWARE DESIGN

The software design of our control system includes two parts: one part is the software design about ’5402, and another is that about PC. The software design about PC consists of the communication and the translation of image format. However, the software design about ’5402 includes the image data receiving, storing, timing for 30 seconds and communication with the PC. As for the software design about ’5402, we should focus on the following aspects: assignment of internal and external memory, CPU initiation, deciding the rates of differentiating of the scanner[24], judging the speed of the moving scanner, sampling and storing of the image data, preliminary handling of the image data and timing for scanner head . In order to further handle the image data with PC, we need to transfer the image data to PC at first. So the software design about PC must begin with communication with ’5402

and receiving the image data, and then we need to extract the image data and establish some format image file.

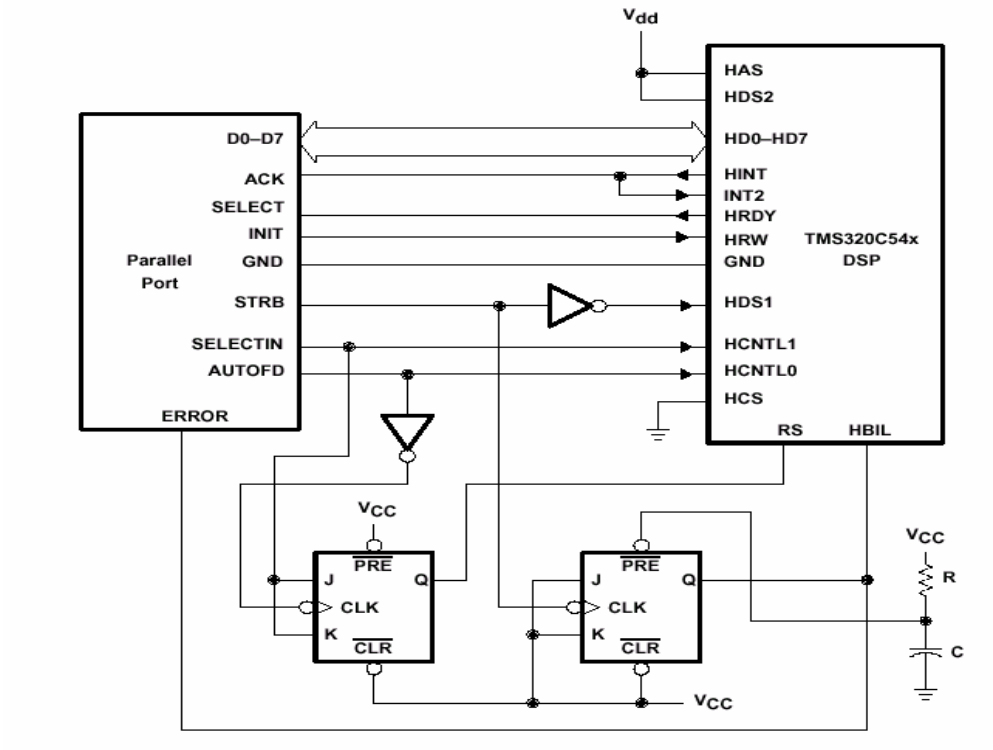


Figure 5 The Interface between HPI and PC Parallel Ports

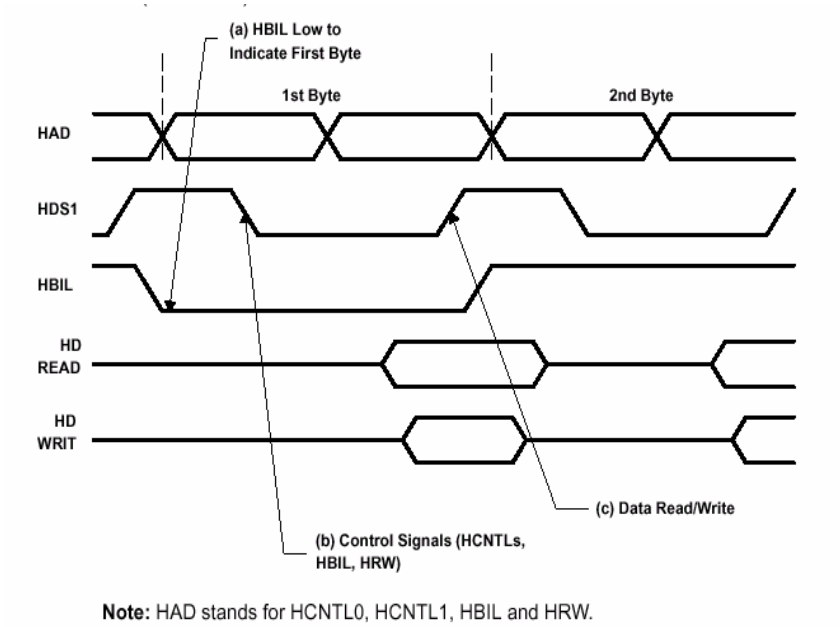


Figure 6 The Time Diagram

6. CONCLUSIONS

The control system based on DSP is designed successfully in lab, and a scanner sample has almost passed all kinds of measurements. The scanner sample handle images as well as the computer interface board. The scanner sample can scan white and black image as 100 dpi, 200 dpi, 300dpi and 400 dpi, which are all clear and real, and they attain our design goal on all technology aspects. But as a product, the control system is not good enough and it needs to be improved further. There are five questions to be resolved:

1) which can s To buy a new scanner head with high 分辨率, can color image. Because '5402 can process color images with high 分辨率 very well.
2) To adopt the small wave algorithm to contract the image data so that the same memory can store more image data.
3) To make full use of the resources of '5402 in order to fasten the sampling rate. For example, to use both McBSP0 and McBSP1 to sample data at the same time, or use the six channels of DMA at the same time to transfer data an so on.

4) To design a USB port to communicate with PC, which is the trend of scanners devlopping.
5) To add some directions on the front side, through which users can know the scanner working status at real time. According to the board design of printer, we can use some colored LBD as direction devices.

7. REFERRENCES

[1] Texas Instruments Inc. TMS320C54x DSP Applications Guide. U.S.A. 2000.
[2] Texas Instruments Inc. TMS320C54x DSP Mnomonic Instruction Set. U.S.A. 2000
[3] Texas Instruments Inc. TMS320C54x DSP CPU and Peripherals. U.S.A. 2000
[4] Texas Instruments Inc. TMS320C54x DSP Algebraic Instruction Set. U.S.A. 2000
[5] Texas Instruments Inc. TMS320C54x DSP.Enhanced Peripherals. U.S.A. 2000
[6] Texas Instruments Inc. TMS320VC5402 Fixed-point Digital Signal Processor.. U.S.A. 2000.
[7] Texas Instruments Inc. The TMS320C54x DSP HPI and PC Parallel Port Interface. U.S.A. 2000.
[8] Shi Yue, Wu Qiong, Chen Jian, An Integrate HPI in DSP. Electron & Acoustics Technology. Vol.1, 1999.
[9] Shi XiaoFeng, Li Jing, Ji ZhiQuan. TheRealization of data transfer by DMA based on McBSP of TMS320VC5403. Microprocessor. May, 2001.
[10] Wang QiHui, Sha ZongXian. A System Which Handles Image Sampling With DSP. Journal of QiQiHaEr Univ. December, 2001
[11] Xu GuoPing, Lou HuiFang. Using VB6.0 To Realize Series Data communication Between PC and DSP. Computer Application. Vol.27, No9, 2001.
[12] Yang lei, Hu XueMing, Yu XingHua. Sample Counting Fee Information from Programmble Exchanger Machine. Modern Computer. Vol.4 1996.
[13] Yang JianDong, Pan LongFa, Ju WenLang A Control System about PC-DSP with High Speed.Small and Micro Computer System. April, 1997.
[14] Liu DongHua, Yi Jun, Liang GuangMing. A System Based on DSP Which can Sample and Handle Data with High Speed. Application of Integrated Circuit. January,2001.
[15] Wang Liming, Zhao YinLiang, Han Yan. The realization of distant data exchange quickly between DSP and PC. Journal of Measure Technology of Hua Northern Industry Institute. February, 2002.
[16] Zhang Li, Deng SiHao. Data Transfer through DMA based on the Multichannel Series Ports of TMS320VC5402. Electron Technology and application. February, 2001.
[17] Liu DongHua, Yi Jun, Liang GuangMing. The Communication Between TMS32032 and PC. Journal of Measurement technology of Hua Northern Industory Institute. February, 2001.
[18] Wang LiuYing. Impulse and Digital Circuit. High Education Publishing Company.
[19] Ma JiaChen, Sun YuDe. Mcs-51 Siglechip Theory and its Interface Technology. Publishing Company of HaErBing Industry University.
[20] Peng QiZong. Applied Teaching Material about TMS320C54X. Publishing Company of Electron Technology University.
[21] Software Design about Data Exchange Between DSP and PC. Microprocessor. May, 2001.
[22] The Realization of Quick Data Exchange Between DSP and PC through EPP. Electron Products World. June A, 2002.
[23] Circuit Design about PC-DSP Communication in a Wareless Identifying System about the Acousic Surface Wave of a Object. Joural of Sensor Technology,2001.

ARQL: An Association Rule Query Language for Association Rule Base

Tang Rongjun, Xiong Zhongyang, Zhang Yufang, Zhang Min
Department of Computer Science, Chongqing University,
Chongqing, 400044, China
E-mail: zyxiong@cqu.edu.cn

ABSTRACT

The emerging data mining system with storage of the mining results naturally leads to the demand of a powerful result query language, on top of which many operations of storing mining results can be easily implemented. This paper puts forward ARQL (Association Rule Query Language) and gives out formal description of ARQL statements. ARQL is a quasi-natural language and can be used to manipulate a kind of stored data mining result. Using SQL in relational database to realize the ARQL statements is also discussed.

Keywords: Association Rule, Data Mining, Query Language, ARQL

1. INTRODUCTION

At present, many data mining systems have been used widely. The quantity of data mining results is becoming more and more. In order to full make use of data mining results, mining results are being stored [1]. Coming with this storage method, how to manage the stored mining results becomes more important. So the operation for the association rules in association rule base have been studied. Referring the design philosophy of the DMQL [3] and SQL, Association Rule Query Language (ARQL) is put forward, formal description of ARQL statements is also given out. ARQL is a quasi-natural language. ARQL Statements will be used to realize the management and query of association rules.

2. DATA MINING SYSTEM WITH STORAGE OF THE ASSOCIATION RULES

A data mining architecture with the storage of association rules is proposed through analyzing existing data mining system architecture [2]. The main idea is to introduce storage system of association rules in the general data-mining system architecture. The mining results are called association rules, which were provided to the users directly in the past, are stored in the rules warehouse through which various operations and management can be carried out. A consistent view of association rules is formed so as to improve the efficiency of sharing and mining association rules. The improved structure solves the existing problem in the traditional data mining system effectively. The mining results, association rules, are stored in the rule base, which supports various efficient operations and managements, is convenient to share association rules among various data mining system efficiently.

3. OPERATIONS OF ARQL

The ARQL provides three kinds of operations for the storage system of association rules: rule base definition operation, rule query operation and rule modification operation.
The rule base definition operation is used to define the structure of a rule base, which stores association rules and drops the rule base when it is no longer used. The rule query operation is used to search the association rules that match the query conditions, which is the most complex operation in ARQL for having the ability to express the complex query conditions. The rule modification operation is used to insert an association rule into rule base and to delete association rules from rule base.

4. SYNTAX OF ARQL

The ARQL adopts an SQL-like syntax to facilitate association rule operations and can be easily integrated with relational query language, SQL.
The ARQL language is defined in an extended BNF grammar, where "[]" represents 0 or one occurrence, "{}" represents 0 or more occurrences, and words in **sans serif** font represent keywords, as shown bellows.
<ARQL>::=
<create_rulebase>|<drop_rulebase>|
<select_rule>|
<insert_rule>|<delete_rule>
Among these definitions, <create_rulebase> and <drop_rulebase> compose rule base definition operation; <select_rule> is rule query operation; while <insert_rule> and <delete_rule> compose rule modification operation. These operations will be expressed as follows.

4.1 Rule Base Definition Operation

4.1.1 The <creat_rulebase> sentence

The <creat_rulebase> sentence is used to create a rule base in which association rules can be stored. Because different association rules have different parameters, the rule base definition operation should have the ability to define corresponding storage structure according to the special association rule.
<create_rulebase**>**::=
createrulebase <rulebase_name>
with body, head datatype <type>[; **parameters** <para_list>]
<para_list>::=
(<parameter_name> **datatype** <type>
{,<parameter_name> **datatype** <type>})
"**createrulebase** <rulebase_name>" defines the name of a rule base. An association rule storage system can define many rule bases for different data mining applications.
The **"with"** statement which contains data type of rule's body, head and parameters describes the structure of rule base.
The "<parameter_list>" defines the structure and data type of association rule's parameters.It should be extendable for different data mining applications' result may have different parameters. The parameter's amount and data type also can be defined.
An example of association rule base definition in supermarket

sales is shown. The name of the rule base is "sales-trend" which stores the association rules mined from a supermarket sales data:
createrulebase sales-trend
with (**body**, **head datatype** char(30);
parameters (support **datatype** float(3,3) ,
confidence **datatype** float(3,3),
time **datatype** date))

4.1.2 The <drop_rulebase> sentence

The <drop_rulebase> sentence is used to remove a rule base from association rule storage system.
<drop_rulebase>::=
droprulebase <rulebase_name>
"**droprulebase** <rulebase_name>" specify the name of rule base which wanted to be removed.

4.2 Rule Query Operation

4.2.1 The "<select_rule>" sentence

The "<select_rule>" is the most important part in ARQL. Through this statement query operation can be executed and the result set of association rules that match the query conditions defined in "**where**" statement will be returned.
<select_rule>::=
selectrule
from <rulebase_name>
[**where** <query_expression>]
[**order by** <parameter_name> **asc** | **desc**]
The "**from** <rulebase_name>" statement directs the query task to a specific rule base "<rulebase_name>".
The "**where** <query_expression>" statement which gives the query condition is optional. If there is no **where** statement, the **selectrule** will return a rule set containing all the association rules in the rule base. For the complexity of the "<query_expression>" statement, it will be discussed separately as follows.
The order by statement, "[**order by** <parameter_name> **asc** | **desc**]", sort the returned rule set by <parameter_name>.

4.2.2 The syntax of <query_expression>

The whole expression of query contains three parts: body query expression, head query expression and parameter query expression.
<query_expression>::=
[**body** <operator> <value_set>,]
[**head** <operator> <value_set>,]
[**parameters** <para_exp>]

<operator>::= **equal** | **include**
<value_set>::=(<value>{**and** <value>}) | (<value>{ **or** <value>})

<para_exp>::=
(<parameter_name><para_operator> <para_value>
{,<parameter_name><para_operator> <para_value>})

<para_operator>::= [!] > | < | = | >= | <= | **between**
The "**body/head** <operator> <value_set>" means the searching process is executed in body/head part of the association rules.
The "<value_set>" defines a condition value set in the rule base. To the association rule, the "<value>" is belonged to the transaction data set. For example, if we want to find the rules which body part include hummer and bike, we can write the "<value_set>" like this: ("hummer" and "bike").
The "<operator>" contains two operators, "**equal**" and "**include**". The "**equal**" means the returned association rules' body/head part must exactly equal to the query condition value set while the "**include**" means the returned rules' body/head part must include the query condition value set.
The "**parameters** <para_exp>" statement defines the query condition to the parameters of association rules. For example, we want to find out the rules which have its parameter "*support*" larger than 69%, we can write the query expression, "**parameters** support>0.69".

4.2.3 Example of a selectrule statement

Here an example of "selectrule" statement based on rule base "sales-trend" will be presented. We want to find out the association rules which have hummer and bike in its body and lock in its head, at the same time, the rules' support count must be larger than 15% and the confidence must be lager than 70%. The result should be listed with the confidence value in descending order. To find out the matching result, the query expression can be written like this:
selectrule
from sales-trend
where body include ("hummer" and "bike"),
head equal ("lock"),
parameters (support > 0.15, confidence > 0.7)
order by confidence **desc**

4.3 Rule Modification Operation

4.3.1 The <insert_rule> statement

The "<insert_rule>" statement is used to insert an association rule into a specified rule base.
<insert_rule>::=
insertrule
into <rulebase_name>
with body <body_set>
head <head_set>
parameters <parameter_set>

<body_set>::=(<value>{,<value>})
<head_set>::=(<value>{,<value>})
<parameter_set>::= (<parameter_name> = <para_value>
{, <parameter_name> = <para_value>})

The "**into** <rulebase_name>" statement specifies rule base which the association rule will be inserted into.
The "**with**" statement describes an association rule which will be inserted into rule base. The "<value>" of body and head is taken from the transaction data set.
The "<parameter_set>" statement gives the parameters set of the rule. The amount and data type of the parameters must be consistent with the rule base definition.
For example, if we want to insert an association rule, "(hummer,bike) ==> (lock), c=80%, s=30%", the insert statement should be written as below:
insertrule
into sales-trend
with body ("hummer","bike")
head ("lock")
parameters (support = 0.3, confidence = 0.8)

4.3.2 <delete_rule> statement

The "<delete_rule>" statement is used to delete an association rule from a specified rule base.
<delete_rule>::=
deleterule
from <rulebase_name>
with body <body_set>

head <head_set>

5. REALIZATION OF ARQL

The realization of the ARQL using SQL is briefly introduced as above. The storage structure of association rule base is consisted of three relational tables [2]. The basic idea is: decomposing an ARQL statement to a series of SQL statements which must be included in a database transaction. Only when the transaction is wholly executed, the ARQL statement is treated as executed successfully.

6. CONCLUSIONS AND FUTURE WORK

In this paper, the association rule query language (ARQL), for managing and utilizing the mining result, is presented. It is convenient to establish the interactive interface of the association rule storage system based on the ARQL. Using the ARQL, many management functions on rule base can be executed, like adding, removing, querying and organizing data, as well as authority management.

There are many issues that needed further studies in this direction, which listed as follows for discussion.

The ARQL in rule base is similar to SQL in relational database, so it is necessary to work out a set of interpretation means to translate ARQL into SQL.

It is necessary to add authority management statement into the ARQL to realize the association rule base.

7. REFERENCES

[1] Parsaye.K. From Data Management to Pattern Management, DM Review, January 1999

[2] Zhongyang Xiong, Research of Storage Method for Association Rules with Relational Algebra, IEEE TENCON'02, 2002

[3] J. Han, Y. Fu, K. Koperski, W. Wang and O. Zaiane. DMQL: A data mining query language for relational databases., 1996 SIGMOD'96 Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'96), Montreal, Canada, 27-34, June 1996

[4] S. Sarawagi, S. Thomas, and R. Agrawal. Integrating association rule mining with relational database systems: Alternatives and implications, In Proc. ACM-SIGMOD, pp 343—354, 1998

[5] Jiawei Han, Micheline Kambr. DATA MINING Concepts and Technique. High Education Press, 2001.5

The Application of XML Technology in E-procurement

Houping , LinJinguo, WangJie
College of Automation, Nanjing University of Technology
Nanjing, Jiangsu, China
E-mail: hp_nj@sohu.com

ABSTRACT

This paper expatiates the process and modes of e-procurement, by comparing e-procurement with the traditional procurement. It introduces XML technology and its application in composing the system of e-procurement.

Keywords: E-procurement B2B E-commerce XML Technology

1. INTRODUCTION

E-procurement is a popular e-commerce mode of B2B. It is also a mature mode of e-commerce. E-procurement is an important part of enterprises realizing e-commerce. XML (eXtensible Markup Language) provides a new idea for data's showing and transporting on the Internet.

2. E-PROCUREMENT

E-procurement and Traditional procurement

The so-called procurement is defined as the systematic activity with the supplying department of enterprises purchasing goods from exterior through all kinds of ways. It is the early stage of enterprises' production and management and has an important influence on the efficiency of enterprises. Procurement takes up the large quantity funds, usually above half of enterprises' grass earnings. Even if the funds of procurement are cost little, it has a big influence on the cost. Therefore, the main path of declining cost and increasing the profits is lowering of procurement cost price.

E-procurement is a kind of purchasing activity of the instant information exchanges and the on-line bargain, which is based on the computer and net technology, e-commerce software, the EDI e-commerce payment tools and the safe system of e-commerce.

Different from the traditional procurement, e-procurement is a type of procurement manner of being based on net technology. The traditional procurement is primarily manual. The communication of both parties is mainly through telephone, fax and visiting. The whole operation flow of procurement has no automation, and the data will be input and affirm repetitiously and continuously. There will unavoidably be errors within the process. Even though purchasing a few and piecemeal goods need complicated and interminable approved-process. The traditional procurement not only prolongs the purchase period, but also consumedly increases the business cost, which may cause the purchase officer's disaffection and to proceed a purchase with the nonstandard steps. In a word, the efficiency of traditional procurement is not high, which causes to add costs and prolong the purchase periods. The traditional procurement neither controls procurement nor has the scale economic effect.

The advantages of e- procurement consists that it can low the cost of management and provide more chances to select production and service, which will reduce risk, shorten period, control the flow of procurement and stock, and make the service of suppliers more perfect. In addition, it can availably provide information of suppliers and lower the price of merchandises. The successful solution of e-procurement can establish the norm purchase process for business enterprise, and benefit to enhance the enterprise's management.

The Analysis of E-procurement Process

The process of e-procurement generally lies in the follows: Bring up the needs of procurement—search for the suppliers and their related information—confirm the selected suppliers —auction on-line—make an order—deliver goods—accept the goods—pay a sum of money—transport to the one providing needs.

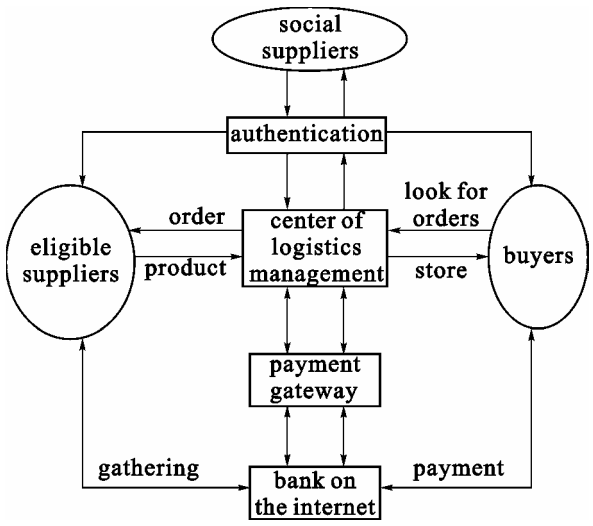


Figure 1 The analysis of e-procurement flow

1) Acquire the purchase information. It primarily comes from the sections of each working and management of stocks. The buyer real-time inputs the information into the databases by web, or each section inputs the need information into databases. In this way, each section can expediently obtain the other party's latest information by web, and the buyer also can on time controls the latest need information towards supplies of each section.

2) Choose supplier and goods. This process takes the form of auction on-line. The buyer can obtain the information of suppliers and goods by two ways. One is the original information of suppliers stored in company, the other is the information searched on the Internet. The former primarily accesses database server, searches supplier and product on the intranet of company. The latter mainly searches for the related suppliers and goods on the Internet, and find out the befitting supplier. Then the buyer will confirm the searched suppliers, ascertain the suppliers who will take part in the auctions on-line. The buyer will organize the conformed suppliers to

auction on line, decide the suppliers and product, finally sign the contract and make orders.
3) Order goods, urge goods and control the process of supplying.
4) Check up the schedule of goods and quality.
5) Accept goods to stores. Input the information of new goods to databases.
6) Payment. The buyer informs the finance section after receiving notify of management section of stores.
This process shows in figure 1.

Modes of E-procurement
The modes of e-procurement decide the choice, usage and development of technology. The process of e-procurement has something with both suppliers and buyers. Especially the information of procurement comes from exterior of enterprises, which provide the possibility of building modes. The modes of procurement mainly include four kinds.

The mode of supplier drive: The mode of supplier drive (Figure 2) points that the supplier puts out the on-line catalogues, introduction and price of product and related information. The buyers obtain information through browse the website of suppliers. They make the last decision by comparing and weighing. They make an order and discuss the payment and related information with the last confirmed suppliers. In this mode, suppliers must plough into large manpower, material and resources to vindicate and renew the catalogue and related information, therefore the cost is higher and operation is complicated. For procurement, they can obtain product information only by browsing the web page. But at the same time, they face the problem that e-procurement can't nicely integrate with the Internal information system of enterprise of backstage. Because the buyer and the supplier communicate through supplier's system, they may take different standards. In this case, the conversion would lower the efficiency of e-procurement and prolong purchase periods.

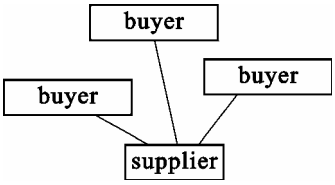


Figure 2 The mode of supplier drive

The mode of buyer drive: The mode of buyer drive (Figure 3) points that the buyer establishes its website on the Internet firstly, then promulgates the information which it needs. The supplier registers its product's information for the buyer review and evaluation. Both parties proceed further communication through the website of the buyer and finish all the procurement. Different from the mode of supplier drive, in this mode, the buyer undertakes to establish, vindicate and renew product's catalogues and related information, which can more closely control the whole purchase process. The buyer can limit the category and specification of product of catalogues, also can set up purchase powers for different workers and quantity restrict.

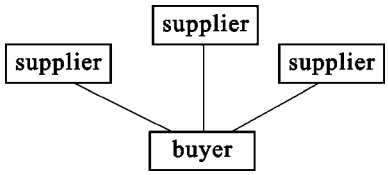


Figure3 The mode of buyer drive

The mode of market: The mode of market (Figure 4) points that suppliers and buyers purchase through the third party. Many buyers and many suppliers have business in common, and every participator has many opportunities. In this mode, whether buyer or supplier, only need announce and describe its own information or require on the third party's website. The third party website is charge of concluding product's information, in order to use of consumer. Like this, the cost for establishing the website is saved, but because this market is the independent third party website, it is very difficult for the market to integrate with the backstage system of buyer. For making up the limitation, some e-markets, especially provided by the e-procurement solutions, all adopt open truss based on XML. This kind of truss gradually becomes the main mode of e-market. Under this kind of truss, the systems can communicate successfully with XML regardless the language of enterprise's taking. At the same time, they provide the integration service of backstage for customer, which makes enterprises to smoothly purchase through the e-market.

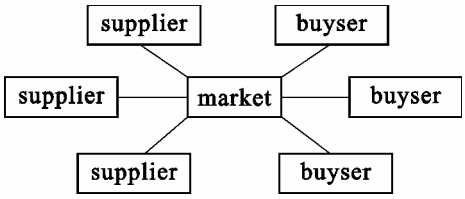


Figure 4 The mode of market

The mode of business community: The mode of business community (Figure 5) points that governments, finance and other parties take part into the market. In this stage, enterprises can join all the trading partners into a general business community through the third party website, which strengthens the market clarity. Once buyer and supplier trade on the same website, the possibility of all kinds of bargains could increase, which will bring more value for all the enterprises that had joined the business community. Government and department concerned establish the laws of e-procurement, which makes the e-procurement have laws to according to. The involving of financing institutions makes e-procurement realize pay by credit or transport account of bank on the Internet and other financial process, which enhances the efficiency of procurement. But in this mode, high technique of information safety, high risk and big invest are needed. This mode need further research.

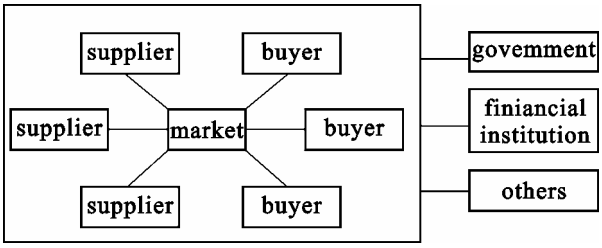


Figure 5 The mode of business community

3. XML TECHNOLOGY

What is XML technology

The XML (eXtensible Markup Language) is a language established by the World Wide Web Consortium (brief name is a W3C) used for describing the organization and arrangement of data text. It is a kind of Markup language, which is similar with the HTML, but XML emphasizes on not how data layout and show in the browser but how to describe the buildup and structure of data convenient for interchanging and organization of data on the net. We know that the same data can be used for different purposes. For example, some firm's sale record data can be used for not only the search of the sale but also analysis and decision-making of the firm's sale. The premise of data being used repeatedly is that data are stored in regular way, according to data's content and independent from application.

Nowadays, the main way of exchanging data on the Internet is browsing all kinds of net pages compile with HTML. In a net page, the contents of data are not separated, but blend with how to be showed. Like this, it is very difficult to withdraw data contents from a web page, which hindering the reuse and exchange of data on the net. In this way, we can see that the HTML is a typical example that contents don't separate from application (application in describe the web page), which causes usage efficiency low. The file of a HTML text includes many data that are only used as showing of main page, and can hardly be used repeatedly by other data user.

The emphasis of XML is organizing and describing data. You can design how data layout in your file with XML. Though the information of structure, different users can acquire data concerned for different purpose. The key of XML success is that it separates data contents from applications in order to enhance efficiency. The database originally being put forward is in order that data are stored in the way independent from the use of data for being used by different applications. We can conclude that XML is same to database, but XML applies in enhancing the reuse and efficiency of data exchange.

Three main factors of XML and forms of text compiling

XML language is an open standard system, which allows suppliers to design marking and property names by themselves according to need. XML primarily contains three main factors: DTD (Document Type Definition, definition of text type) or XML Schema (XML mode), XSL (the eXtensible Style Language) or Xlink (eXtensible Link Language). The DTD and schema are used for stipulating the logic of document, and defining elements, properties of elements and relationship. It can help analysis procedure and verify the legitimacy of markup of XML documents. The XSL is used for expressing the form of XML texts, which can make web browser change the style of documents at the customer and provide formatting makeup functions when XML document showing and printing. The Xlink further expands the hypertext link and affixture functions for link. It has expanded link on the web from simple link to bidirectional and mulriple link and allows customers append instructions for hypertext link. The following will give an example to explain the forms of XML documents. Supposing, a customer purchases 2000 pieces of 3.5in disks at price of 4.0 yuan by the accounts of hoping on the Internet on the 3rd of March 2002. The following shows the order being compiled with XML.

```
<?Xlm version="1.0" standalone="yes"
encoding="UTF-8" ?>
<! DOCTYPE purchase SYSTEM " purchase.dtd">
<purchase>
<date>3/3/2002</date>
<account>R6-Rorbertson</account>
<item>
<itemdes>3 1/2 Fluppy Disk</itemdes>
<quantity>2000</quantity>
<unitcost>4.0</unitcost>
</item>
</purchase>
```

The merits of XML technology

The main four characteristics of XML technology is good data memory format, expansible, highly structured, and easy to transport by Internet, which results in prominent performance. Because XML technology can define it's markup language aiming at specifically application. And this characteristic make XML apply in e-procurement, which providing all kinds of independent solutions, according to different systems. The merits of XML lie in the follows.

1) It is easy and effective for programming with XML technology. The syntax of XML is similar to HTML and it is very easy to deal with documents including of XML.

2) XML technology is the bridge among computers. Computers can process bi-directional communication with XML technology, which is the most important for the application in e-procurement. The author of web page layout who may map out and design file formats with XML technology, and needn't take up the fixed HTML format can have more abundant syntax and more functional standard language.

3) XML technology will accelerate the development of e-procurement. The characteristic of easily exchanging and transmitting data is very important for e-procurement.

4. THE APPLICATION OF XML TECHNOLOGY IN E-PROCUREMENT

The emergence of XML technology provides a new way for data express and deliver on the Internet, which makes B2B e-commerce based on the Internet have new opportunities. In the applications of e-procurement, the main function of XML is that all the exchanging things express, store and exchange. It also includes of recording and configuring data of company or personal consumer. The agility and extend of XML benefits to realization of e-procurement. The open and based on document XML is very fit for exchanging information through servers.

Not long ago, the non-payoff business association of Commerce Net consisting of 500 members suggested describing product, software of service catalog, business rules and system data with XML. The criterion of Commerce Core will define how to give the company's name, address, price, item, quantity, and so on to marking. We believe that it can make the data of enterprises exchange simply and smoothly. But presently, the main suppliers of software such as Ariba, Commerce One, Microsoft all have their own standard. The united standard and norm need much more time to operate. We believe that it would be realized in future.

XML technology is very important for the development of e-procurement. The development of e-procurement mainly introduces the second generational web technology with XML as representation. The process of e-procurement is a kind of electronic fashion, which integrates the exchange of information and bargaining actions among the purchase, the supplier and the middleman who serve for the purchase and the supplier, such as finance organization.

The e-procurement based on the Internet is different from the traditional EDI applying in private net or increment net. The most primary reason is that the little scope, partial, single and costly commerce concept is spread to the exoteric, public, compatible and low-cost system. In order to adapt to the change, it must be provided with some base that realization e-procurement on the Internet. The immanent need mainly represent the standard of information and technology of data integration within net deal. XML technology can settle the difficult problem very well. The main points lie in the follows:

1). XML technology possesses some good characteristics, which makes it be the international language of e-procurement. XML technology is based on the Internet communication protocols such as HTTP. Not only the deploy method of XML technology is easier, but also the cost of it is much more lower. So XML technology is the best method for the middle and small enterprises deploying EDI without information departments.

2) XML technology will change the communication method among enterprises, and become the actual criterion of enterprises' data integration. Along with the e-commerce development among enterprises, it is needed for describing and exchanging information about purchase orders, components and commodities.

3) XML technology will be able to replace the traditional EDI system, and the small company with XML also can carry through e-procurement by sending purchase orders. Using XML technology will change traditional value chain, thereby realizing management of supply chain.

4) Connect of different systems is very important for realizing e-procurement, and the integration of enterprises' systems is the most important problem when carrying through uniting and annexing among enterprises. The immediate advantage of e-procurement based on XML technology is that it makes new enterprises and little enterprises take up e-procurement. Because of convenient application, broad usability and lower cost, the enterprises making use of traditional EDI should also begin to exchange data with XML technology in order to take up more copartners and win more profit.

5. REFERENCES

[1] 江苏省计划经济委员会.采购管理大有潜力[J]. 江苏:经营与管理,2000,(6)

[2] Detourn N. 2000. B2B E-Commerce the dawning of a trillion-dollar industry. Motley Fool Research, March 14, http://www.FoolMart.com (Last accessed July 15, 2000).

[3] Dennis P Geller & Bradley L.Hecht Electronic Procurement: the Extranet and You [J]. Intranet Design Magazine, December 1999.

[4] Elliotte Rusty Harold 著.XML 实用教程.机械工业出版社

[5] SimonSt.Laurent.WhyXML. http://members.aol.com/SimonSTL/xmo/whyxml.htm

[6] 杨利.B2B.电子商务—网络经济的下一个浪潮.计算机世界:产品与技术 2000(4)第 15 期 C 版

[7] Amstel P. An interchange Format for Cross media Personalized Publishing [J]. Computer Networks, 2000(33): 179-195

[8] Salvato G. Presentation and Exchange of Business Models with CIMOSA XML [J]. Computers in Industry, 1999(40): 125-139

[9] 查理·马丁 .用 XML 组建电子商务系统[M]. 北京:北京希望电子出版社,2001,5

[10] Sean McGrath XML 应用实例 建立电子商务应用[M] 北京:人民邮电出版社, 2000, 10 [5]

QoS Implementation Based on IntServ and DiffServ in Linux

Xu Yi Gui Ruifeng Li Layuan
Wuhan University of Technology
Wuhan,Hubei, 430063, P.R. China
E-mail: xuyi@mail.whut.edu.cn

ABSTRACT

With the increasing development of Internet, such as multimedia, the desire for Quality of Service becomes higher and higher. There have been several proposals for providing Quality of Service in IP network. Among them the most important is the Integrated Services (IntServ) and the Differentiated Services (DiffServ). This paper provides an overview of the Quality of Service support in Linux, and describes the implementation of both of Services.

Keywords: QoS, Linux, traffic, DiffServ, IntServ, Services.

1. INTRODUCTION

As with many research and development projects especially in the Linux area, this work is not ready but in progress. It brings up new aspects of Quality of Service support that fits the user's needs and tailors much better to individual user preferences. Linux is a stable and effective operating system and it represents a strong competition to other UNIX versions especially to Windows NT in the small to medium server market. One of Linux main advantages is the great flexibility of its networking code. A Linux box can be used as a gateway, router, or firewall at the same time and beginning with the kernel versions 2.1.x. with the introduction of traffic control, Linux also has support for QoS algorithms. Consequently IP QoS architectures, Integrated Services and Differentiated Services are supported for Linux. This paper discusses the QoS support that is available in the recent Linux kernels. The QoS support in the kernel provides the framework for the implementation of various IP QoS technologies, like integrated services and differentiated services.

2. SUPPORT FOR QoS IN LINUX

Linux, a shareware operating system, has support for a number of advanced networking features. Besides the reliable TCP/IP protocol suite, a number of new features like firewalls, QoS, tunneling etc, have been added to the networking kernel. These advanced networking features have been implemented in the Linux kernel, from a configuration, implementation and usage standpoint. Examples of usage and pointers to references have been given when appropriate. The advanced

networking features that have been dealt with in this document include the Quality of Service support in Linux, which encompasses a description of the differentiated Services and Integrated Services.

2.1 QoS Basic Principle

The basic principle involved in the implementation of QoS in Linux is shown in Figure 1. This figure shows how the kernel processes incoming packets. A router forwards received packets directly to the network, e.g. to another interface (1). If the node is also an end system (server, workstation etc.) or an application level gateway, the packets destined to it are passed to higher layers of the protocol stack for further processing (2). This can also include manipulation of fields and then forwarding to the network (3) again. An end system can generate packets by itself, which then will be sent through the protocol stack to the forwarding block (4). The forwarding component does not only include the selection of the output interface but also the selection of the next hop, encapsulation, etc. From there, the packet is queued for the particular interface. This is the point of traffic control execution. Manipulation, such as delaying packets, changing header fields, dropping etc, can be done there. After traffic control has released the packet, the particular network device can pick it up for transmission. The output queuing block is triggered by the output interface. For processing the next packet, the interface sends a start signal to the output queuing block. After compilation of TC and load the modules, the code components can be added via the command line or a management tool (a Shell or Perl script) to the outgoing queuing block [1]. The code consists of queuing disciplines, classes (the identification of a queuing discipline), filters, and policing functions (within filters and classes). Packets, which are forwarded over the same interface, may desire different treatment. They have to be enqueued into different queuing disciplines. For an enqueued packet the called queuing discipline runs one filter after the other until there is a match with a class. Otherwise, the default queuing discipline is used. In the case of a match, the packet is enqueued in the queuing discipline related to the class for further manipulation of the packet. Different filters can point to the same class. Policing functions are required in the queuing disciplines to ensure that traffic does not exceed certain bounds. For example, for a new packet to be enqueued, the policing component can decide to drop the currently processed packet or it can refuse the queuing of the new one. Each network device has an associated queuing discipline, in which the packets are stored in the order they have been enqueued. The packets are taken from the queue as fast as the device can transmit them.

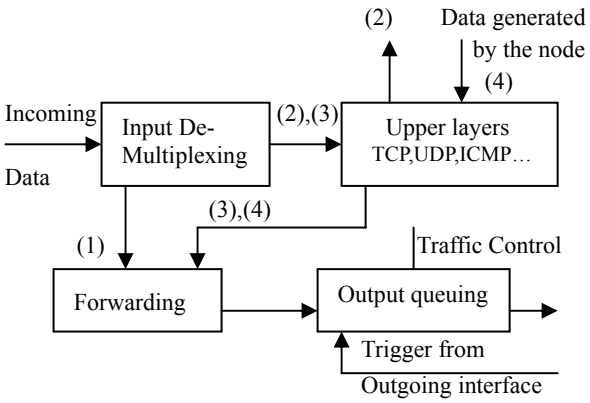


Figure 1 Linux Traffic Control

From an implementation standpoint, When queuing disciplines are created for a device, a pointer to the queue is maintained in the device structure. The IP layer, after adding the necessary header information to a packet, calls the function dev queue xmit. A portion of this code is shown below.

```
q = dev->qdisc;
if (q->enqueue) {
      q->enqueue(skb, q);
      qdisc_wakeup(dev);
      return 0;
}
…
if (dev->hard_start_xmit(skb, dev) == 0)
…
```

This function shows that before actually sending the packet on the output interface (by doing a hard_start_xmit), the packet is enqueued in the queue maintained by the device, if one exists [2]. Thus, a mentioned before, traffic control is implemented just before the packet is sent to the device driver. As already mentioned, the Linux traffic control mechanism provides the basic framework for the development of integrated services and differentiated services support in Linux.

2.2 QoS Architecture

The overall picture the QoS architecture is given in Figure 2. Basically, it consists of a set of Monitoring & Control entities, which are each associated with a resource. The resources currently under consideration in the end system are the processing time, the available data rate shared between different flows and the available network links in a multi-homed end system [3]. The Monitoring & Control entities monitor the demands. The control part realizes the interaction with Monitoring & Control entities. They accept demands for a modified resource share, and pass this demand down to another entity according to some adaptation and coordination rules. Many application- oriented approaches try to maintain information about the application's requirements and perform resource reservation in the end system based on a head calculation of needs. Opposed to this, our QoS architecture is strictly user oriented. The resource sharing in the end system and the bias of the resource distribution should be done in favor of those applications the user is currently focusing on. Different parts of the architecture have been realized up, and the network switching is part of the current work. Whereas application adaptation is a well-known concept in distributed multimedia systems, we first introduced the notion of proactive applications. In contrast to adaptive applications that try to adapt to the current situation but cannot actively control resources, proactive applications actively take effect on the sharing of resources in advance. Thus, beside user interaction by means of the Q-Button, applications can interact with the system on their own.

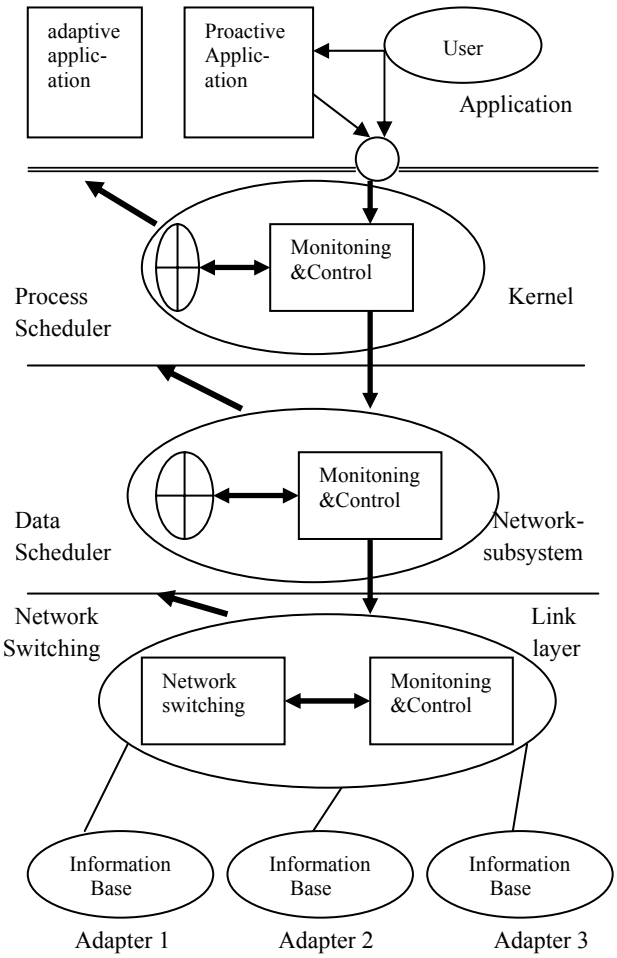


Figure 2 QoS architecture

3. DIFFSERV AND INTSERV IMPLEMENTATION QOS

3.1 Integrated Services on Linux

The Integrated Services implementation in Linux is achieved using a special RSVP daemon. This daemon allows workstations to work as end systems and routes, and has been ported to others platforms. The RSVP daemon has three major interfaces, which provides support for the different roles of an IntServ node, an application interface, a traffic-control kernel interface and a routing interface. The application interface is used to provide a QoS support, or the development of applications. Using the RAPI, the applications can specify their QoS requirements, and use the RSVP daemon to provide the signaling mechanism to reserve the necessary resources in the network. This API has been modified to match the RSVP that is being proposed as a standard interface. The traffic-control kernel interface enables a node to act as the network element in an Integrated Services network. The interface is used to pass the reservation parameters to the traffic control (TC) functions in the kernel [4], via a kernel-specific TC adaptation module specially implemented for Linux. Thus, currently, Linux can be used as an Integrated Services network element. The IntServ implementation in Linux provides a framework for applications with special requirements to reserve resources in the network. The reservation is specified by each application using the RAPI, with parameters configured in compliance with the necessity of each one. This enables a better network resource allocation and QoS specification than in the Differentiated architecture.

3.2 Implementing an RSVP Classifier Module in the Linux Kernel

Nowadays Linux is the only operating system that is equipped with efficient IntServ support. There are sophisticated lower layer filtering and scheduling mechanisms implemented in recent Linux kernels. However, even Linux lacks an RSVP Classifier module that is responsible for forcing QoS packets on routes selected by QoS routing. Currently routing is done based on a common kernel routing table. Incoming packets, regardless of their QoS or best-effort nature are forwarded in the same manner. The best match is searched in the routing table according to the destination address of the packet. Packets are then matched against RSVP filters, and inserted

into the appropriate scheduler's queue. Nevertheless, in the previous section we concluded that the kernel routing table does not lend itself to install QoS routes easily, thus some modifications are required to the current routing infrastructure. Here is a brief outline of the structure of an RSVP compliant Classifier. Although there is no such module currently available, the basic building blocks (i.e. RSVP filtering mechanisms, Net link Socket architecture, Net filter) are implemented in recent kernels. The main components of the RSVP Classifier are the packet filtering blocks, which filters all incoming packets by matching them against subsequent RSVP filters. A match indicates that the packet belongs to a known session, and there must be a QoS routing entry and a queue associated with it. The packet is then taken away from the standard Linux or warding engine, and the appropriate QoS routing entry is looked up. This specifies the gateway address of the packet (next-hop's IP address) and the prioritized queue, it has to be inserted in. When a reservation is established, RSVP installs a filter and the queuing discipline for the new reservation in the kernel, adds the proper QoS routing entry [5]. Detailed implementation issues of the RSVP Classifier are beyond the scope of this paper. Also it is important to note, that proper user domain applications have to be implemented to assure uniform access to lower layer QoS routing mechanisms.

3.3 Differentiated services on Linux

A lot of applications nowadays use real time data and mission critical data. These data require a guarantee, in terms of Bandwidth, latency and other data transfer parameters. IP does not provide this kind of guarantee. It has one class of service: the best effort service. There are many architectures introduced that provide QoS. Some of the more prominent models are label switching, and integrated services/RSVP. The label switching model includes, frame relay, MPLS and ATM. In this model path forwarding state and traffic management or QoS state is established for traffic streams on each hop along a network path. This model permits finer granularity resource allocation to traffic streams. But this increased granularity comes at the cost of additional management and configuration requirements to establish and maintain the label switched paths. While in the integrated services/RSVP model, traditional data gram forwarding is used as the default case, but allows sources and receivers to exchange signaling messages, which establish additional packet classification, and forwarding state on each node along the path between them. In the absence of state aggregation the amount of state on each node scales in the proportion to the number of concurrent reservations, which can be potentially large on high-speed links. This model also requires application support for the RSVP signaling protocol. The Differentiated Serviced implementation on Linux has its main characteristics flexibility and simplicity. The simplicity is a result of the Linux platform that has already many of the functions required for the Differentiated Serviced nodes implemented. The flexibility is a result of the design of the implementation, which facilitates nodes configuration and experimentation with PHBs. It is easy to notice that the Linux traffic control components can be used to implement a great deal of the functionality of a Differentiated Services node. In fact, most of the DiffServ implementation is the components already available in the Linux kernel. A Linux node in a Diffserv domain can act as either an interior node or a border node. For an interior node, the Linux Diffserv implementation has two different mechanisms: a BA classifier, and the Per-hop-behaviors. The BA classifier is used to separate the packets in compliance with their DS code points. This is implemented using a new type of queuing discipline created for the Diffserv support, and a new type of filter. The BA classification is performed by first retrieving the content of the decode point of each packet, using the queuing discipline scheme mark. This information is then stored in a field associated with each packet. After the filter class TC index is used to classify the packets in different classes, using the information in skb->TC index [4]. Finally, the packets are sent to the corresponding internal classes, which implement the corresponding PHB.

3.4 Usage of Linux QoS Package Component

The Linux operating system is distributed with an experimental QoS package that offers a simple means of providing QoS based routing. Through using a Linux workstation as a routing gateway, it is possible to perform packet routing and queuing through the network device(s) of the Linux router. The QoS support provided is even capable of offering Differentiated Service support, where packets routed through the router can be marked as belonging to specific QoS based classes in their packet headers. This offers the ability for a simple DiffServ domain to be implemented with Linux routers acting as entry and exit points to a core network. The Linux QoS implementation is based on queuing The Linux QoS implementation is based on queuing disciplines (qdisc), classes and filtering components [6]. The packets of a particular network interface can be assigned to a qdisc where they can be filtered to QoS classes. The packets are then transmitted on the network based on their assigned priority and bandwidth requirements. The Linux package has been utilized in this design to provide a QoS mechanism as shown in Figure 3. The Linux QoS Manager object in the design performs the necessary setup of the Linux routers to establish QoS based routing pathways.

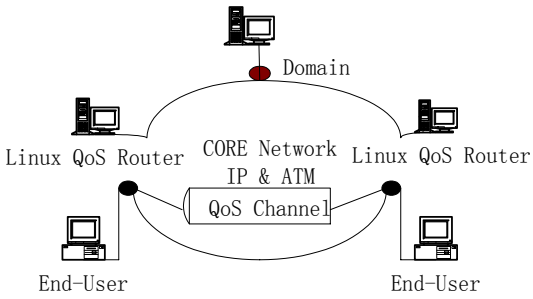


Figure 3 Linux QoS package in providing QoS support

As is shown in figure 3, this arrangement together with the use of ATM connections allows for a transparent network that is capable of offering a degree of QoS based connectivity between the edges of the core network. In the current design, the Linux routers are setup to assign a particular QoS based class to traffic based on the IP header addresses and port numbers of the packets.

3.5 A path migration from IntServ to DiffServ

The scenario we have outlined here is made of a number of ISPs' backbone where the DiffServ model could achieve a flavor of QoS and Campus or Corporate Networks where the Integrated Services model can be used to a different flavor of QoS. The two solutions are not interchangeable because of the IntServ's serious scalability problems we have seen, but on the other hand, DiffServ has administration problems in the allocation and accounting of the resources needed by applications. We can state that there are many pieces in the QoS puzzle that are already in the right place, while other

pieces have to be better positioned. A new IEFT proposal [PROXY] able to leverage both of the models we have just described, is going to further accelerate the deployment of RSVP applications and DiffServ mechanisms so that there will be room for everybody. The better description of this new proposal comes from an economic viewpoint: there are some transit networks (ISP: DiffServ Domains) that can sell services to some stub networks (Corporate/Campus Networks), which in turn provide services to their end systems (users/applications). In other words we have RSVP in the stub network, where network managers have to deal with single user applications, DiffServ in the Transit Network to cope with the scalability issue, and a new element placed at the edge between the Transit and Stub network acting as an IntServ/DiffServ proxy. This proxy is a router able to treat RSVP requests as a small number of aggregates, apply Admission Control based on resource availability within the DiffServ domain, shape outgoing packets and mark each packet with the set of DSCPs (DiffServ Code Points) used for the mapping between Integrated Services and the services available in the DiffServ domain [7]. This model allows a smooth deployment of QoS and above all separates Campus needs from ISP needs.

4. VICE MAPPING IMPLEMENTATION IN LINUX

The Intserv to Diffserv Service mapping has to be implemented. This means that based on RESV Flow spec parameters an appropriate classifier should be set up. However, due to the problems with the RSVP daemon the service mapping could not have been implemented as desired. Without the admission control on the RSVP/Intserv flows first there is no knowledge whatsoever about the available resources. Secondly the processing of the Flow spec parameters necessary to implement service mapping would have been implemented as part of the Linux specific traffic control files, which were not used. Therefore, in order to have at least a basic working prototype of the RID border router the following was done. The Diffserv domain administrator will configure the RID beforehand appropriately to conform to the agreed SLS. This administrator will also configure the Multi Field classifier based on the static SLS that the Diffserv has negotiated with its Intserv customer. The Multi Field classifier will filter the packets based on IP header fields, i.e. the IP source address, IP destination address and destination port number. The classifier used for this purpose is u32. What this means is that the Intserv customer flows will be classified based on their IP header fields. The Multi Field Classifier is installed dynamically upon “acceptance” of the RESV message and deleted upon “release” of the resources, i.e. upon receiving of RESV Tear message [8]. This was implemented by just simply making a system call to the executable TC scripts written for this purpose. The system calls are made in the RSVP llkern.c file as shown below:

```
#ifdef SCHEDULE
handle = TC_AddFlowspec(kp->tcs_OIf, rp->rs_spec,
&Path_Te, adspecp, TC_kflags, &Fwd_specp);
kp->tcs_rhandle = handle;
system("/local/QoS/scripts/correct/filterbuffy.eth1");
#endif
```

and

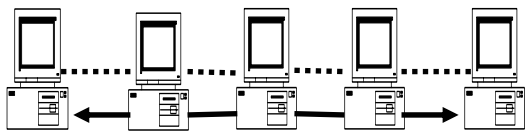
```
#ifdef SCHEDULE
TC_DelFlowspec(kp->tcs_OIf, kp->tcs_rhandle);
system("/local/QoS/scripts/correct/delbuffy.eth1");
#endif
```

Further, since RTAP was used for initiating RSVP sessions between the hosts, there is no possibility to create several RSVP session at the same time for e.g. between the same pair of hosts on different port numbers. With RTAP, there can be only one RSVP session at a time, there was no possibility to have multiple hosts connected to the same router. Thus, what can be derived from above is that this completed implementation of the RID router proto type is very basic. It is to be seen only as the implementation of the RID router concept and not as a real RID router implementation.

5. THE RSVP/Intserv AND Diffserv SIMULATION MPLEMENTATION

5 .1 The RSVP/Intserv and Diffserv Framework

For implementing and testing the RID router the RSVP/Intserv and Diffserv test bed was set up as depicted in Figure 4. Figure 4 RSVP/Intserv and Diffserv WMR Test Bed. All five PC are running Linux Operating System, Redhat 7.2 respectively, kernel versions 2.4.7. Besides Buffy all the machines are Diffserv enabled and apart from spike all the machines are RSVP enabled, i.e. the RSVP daemon is compiled and running. RTAP is used for generating RSVP/Intserv flows and a TCP for generating UDP traffic. There is also a simple generator used for generating background traffic. The configuration of the PC is as hosts and routers. Buffy and Marcie are configured as RSVP aware hosts, which initially use RTAP for creating, modifying or releasing RSVP sessions. They belong to two different subnets that are configured to play the role of RSVP/Intserv access networks. Willow and Tara are configured as RID border routers and are running the modified version of the RSVP daemon in order to handle the Intserv/Diffserv interoperability. They have one of the interfaces configured as Intserv and the interfaces towards Diffserv are configured to handle traffic aggregates. The Diffserv will perform policy, shaping, dropping and remarking on incoming packets according to the agreed SLSs and in this way it will protect the Diffserv domain. Appropriate behavior is configured by means of the TC shell scripts, which are installed before the RSVP daemon initination. Further, both Willow and Tara will encapsulate RSVP packets into UDP on the interfaces facing the Diffserv side. For this purpose the RSVP daemon configuration file will have the function next to the interface. Spike is configured as Diffserv core router, such that it checks the DS field of the incoming packets in order to retrieve the DSCP byte and assign them to the appropriate class [8]. The packets with no DSCP set are to be assigned to the best effort class. So the encapsulated RSVP packets will be treated by spike as best effort.



RSVP/Intserv—Diffserv traffic

Background traffic

Figure 4 RSVP/Intserv and Diffserv WMR Framework

5.2 Static SLS between Intserv and Diffserv

Initially the service mapping of the Controlled Load to AF PHB is to be performed. In the test bed, the Diffserv is configured to support two AF classes (AF1 and AF2) with three-drop priority levels. The Diffserv domain in the test bed can support maximum 10Mbit bandwidth. This bandwidth is divided between the Best Effort and AF classes. The support for AF1, AF2 and BE is implemented by means of CBQ, while AF1 and AF2 support is implemented by means of GRED with assigning GRIO, priority levels to the VQ. On the Intserv side the CBQ is configured in order to handle RSVP/Intserv traffic although there is no traffic control interface modules in the RSVP daemon to use it. This bandwidth shared class should be the same. Each is assigned 5Mbit/s. Further the AF class bandwidth should be divided between the different classes. The AF1 class has 3 Mbit/s assigned and a higher priority and AF2 has 2Mbit/s assigned and a lower priority. The customer (e.g. Buffy) in this SLS requires different drop priorities for the traffic generated to the same destination address on different port numbers [8]. This is depicted in Table 1.

Table 1 Static SLS parameters negotiated between Intserv and Diffserv

Reserved Class bandwidth	AF Class bandwidth	CL Tspec parameters and destination port numbers		
5Mbit/s	AF 1.1[0x0a]	3M bit/s	[CL: 1280 256 64 1280 128]	5000
	AF 1.2[0x0c]		[CL: 640 256 64 1280 128]	5001
	AF 1.3[0x0e]		[CL: 512 256 64 1280 128]	5002
	AF 2.1[0x12]	2M bit/s	[CL: 384 256 64 1280 128]	5003
	AF 2.2[0x14]		[CL: 256 256 64 1280 128]	5004
	AF 2.3[0x16]		[CL: 128 256 64 1280 128]	5005

The association of the controlled load flows (CL) to the AF PHB is based on the type of applications to be used. It is said that the packets with larger token rate b/r , i.e. high burst flows will experience more than the less burst flows. The real-time applications, such as e.g. VoIP do not generate burst traffic, while the other applications such as for e.g. web browsing are highly burst applications. The real-time applications in terms of AF PHB will be assigned to the highest level-class, e.g. AF 1.1. Therefore the non-burst Controlled Load flows with the smallest b/r ratio such as [CL: 1280 256 64 1280 128] will be mapped to the AF1.1 And the high burst flows such as CL: 128 256 64 1280 128] will be mapped to AF 2.3.

6. CLUSION

Technology aimed at Quality of Service deployment is quite mature, but we are not sure people have completely learnt the lesson of these early years of QoS experiments [9]. Some questions arise spontaneously and lead to some controversial answers. QoS in the Internet, up to now it has been a problem of political agreement. Congestion and QoS for an Internet Service Provider, they are both necessary in order to get rid of debts. It's not clear whether it is needed or not. With the fast development of high-speed backbone network and tremendous traffic brought by the newly emerged multimedia communication, more and more Internet traffic is pushed to edge equipments and end hosts where application servers are located [10]. The bottleneck in end hosts caused by this phenomenon triggers the necessity to improve the intelligence and efficiency of end systems. Furthermore, the multimedia traffic needs End-to-End QoS guarantee instead of simple backbone network QoS.

7. REFERENCES

[1] Torsten Braun, Hans Joachim Einsiedler1, Matthias Scheidegger, Günther Stattenberger. A Linux Implementation of a Differentiated Services Router. http://citeseer.nj.nec.com/a-linux implementation of.html

[2] Savanan Radhakrishnan(1999). Linux -Advaced Networking Overview Version 1

[3] Marc Bechler and Hartmut Ritter. QoS in the Linux Operating .SystemTechnical .Report.http://citeseer.nj.nec.com/qos in the linux.html.

[4] M.F.P.Santiago and M.A.RDantas. Quality of service on Linux. http://citeseer.nj.nec.com/qos.html

[5] Gábor Rétvári. Issues on QoS based Routing in the Integrated.Services.Internet.http://citeseer.nj.nec.com/I-ssues-on-qos-based.html

[6] B. Lai, H.E. Hanrahan. The Design of a TINA based Stream .Management/Binding.Framework.http://citeseer.nj.nec.com/laioodesign.html

[7] Nicola Chiminelli CSELTvia G. Reiss Romoli, 27410148 Torino .Architectural requirements for IP end-to-end.QoS. http://citeseer.nj.nec.com/Architectural requirements for IP end-to-end QoS.html

[8] Interoperability of Integrated Services and Differentiated .Services .Architectures.http://citeseer.nj.nec.com/interoperability of integrated -Services.html

[9] IEITF-DiffServ-Working Group,"Differen tiated Services for the Internet." http://www.ieff.org/html.charters/diffserv-charter.html

[10] Xiao and Ni,1999] Xipeng Xiao e Lionel M.Ni(1999).Internet QoS: The Big Picture.IEEE Communications Magazine

Middleware for the Micro-Option

Wang Ke and Wang Qianping
Department Computer Science and Technology
of China University of Mining Technology
Xuzhou City, JiangSu Province
E-mail: wangke@cumt.edu.cn, qpwang@cumt.edu.cn

ABSTRACT

Micro-Option is a method for optimal selection and atomic reservation of distributed resources in a free market environment. This method involves two aspects. One is an optimization problem that is dependent on the way one chooses to model the resources. The other is central to lease-based market. MMM(middle for method management) is an infrastructure for managing the deployment, integration, distribution, and use of application services via World Wide Web. The MMM can be used to implement software leasing over the internet. So we discuss to solve the second problem of the Micro-Option through the MMM in this paper.

1. INTRODUCTION

Many business processes can be described as orderings of tasks. The problem in supply-chain management is to discover resources that can be used in business processes, select the best collection of resources for the task form the set of resources that are discovered, and then secure commitments for all the resources required for the task .The "best" resources are determined by an objective function associated with the process. For example, the objective function may give greater weight to resistance to temperature function may give greater weight to resistance to temperature fluctuations than to size. We consider the problems where commitments for all the resources required for the process must be atomic in the sense that all the resources required to execute the process must be obtained or nine of them should be obtained. The main problem we introduce and solve in this paper is:

Given a partially-ordered set of tasks that involves resources such as people, machines and information; and given an objective function, select the set of resource instances to perform the tasks that maximizes the objective function and reserve the resources in set atomically, committing each resource instance to the negotiated start time, duration and price.

We believe that many resources will soon be traded in lease-based markets. Lease-exchanges will become common for leasing goods and services for specified periods of time. In this paper we identify and study one of the central problems in lease-based markets.

To make the problem more concrete, consider the following simple process :A traveler named Jane is planning a trip from Los Angeles to Paris and needs to reserve round-trip airline seats ,a hotel and a car in Paris. She constrains her trip to take place in a 5-day time frame. Furthermore, she defines an objective function F that depends on the quality of airplane seats, hotel, rental car and cost .Jane's problem is to find a reservation plan consisting of tickets, a hotel and maximizes her objective function, satisfies the constraints, and that can be scheduled and reserved with the least possible overhead cost.

Important properties of this problem are:

(1) The resources are physically distributed and are controlled by different organizations.

(2) Resource availabilities may change often and in arbitrary ways .For instance , the availability of a car in Paris has no direct relationship to the availability of a seat on a United Airlines flight.

(3) All the resources need to be reserved atomically, there is no point in reserving a hotel in Paris if Jane cannot get to Paris

(4) Jane must lease resources in a temporal dependency specified as a partial order.

To solve her problem, we decompose it into the following two problems:1)how to select the best resources that, when reserved, would maximize Jane's objective function; 2)how to actually make the reservation atomically with the least cost overhead.

The first problem is an optimization problem that is dependent on the way one chooses to model the resources. The algorithm is given in [1]. We would not discuss it here.

The second problem is central to lease-based market. An atomic reservation is complex, because suppliers and consumers are distributed and the information they have about each other is always out date. At any time, a consumer only knows some old information about a resource, which may have changed since it was acquired. Thus, a brute-force concurrent reservation of a set of resources has no safeguard against failure. One might recognize this problem as a distributed atomic transaction problem, where all the resources are asked to agree to do their tasks under the specified condition atomically.

All the distributed resources could be reserved on a software provider platform, managed though the MMM middleware. The MMM would implement the Micro-Option contract.

Next section we present the basic idea of atomic reservation. In Section 3 we describe the MMM infrastructure. We give the implementation of the MMM for the Micro-Option.

2. THE BASIC IDEA OF ATOMIC RESERVATION

we are now commencing an atomic distributed reservation transaction between a consumer and all the resources chosen for one of the complete reservation between a consumer and all the resources chosen for one of he complete reservation plans, while maximizing the total objective function. The goal is to make a reservation of at least one of these plans, preferably the one with the highest objective function value.

The information about the availability of the resources in the plans may be out of date (because of network delays, followed by the feasibility and optimality calculations). Some of the resources may have been leased-our for the periods of time the optimization stage has scheduled them. If the consumer attempts to make a reservation of such a plan, the attempt will fail. it is undesirable to start reserving a plan and failing some time in the process, because the consumer spends the full reservation cost for each re source it has already reserved .Each time a reservation fails the total reservation costs go up. To improve the chances of a successful reservation, one needs a mechanism that would give the

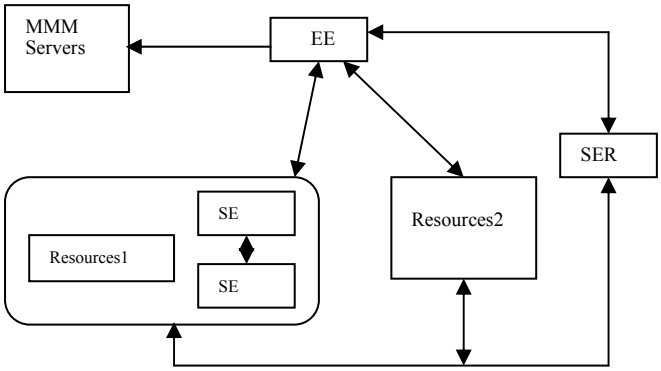
consumer insurance that 1)the time negotiating a reservation agreement is not wasted (because wasted time decreases the quality of the knowledge about resource availability) 2)the costs of making the reservation are as small as possible.
Any negotiation process between entities in a free market has an opportunity cost associated with it, because with any strategy the resources would have to 'hold' some of their time slots open while the negotiation takes place. These slots are the commodities for sale and the opportunity cost is the cost of holding them open. Either the consumer or the resource needs to incur this cost .
Definition :Micro-Option is a short-term right that can be purchased by a consumer to reserve a resource at some specific time in the future for a specific duration and price .

3. THE MMM ARCHITECTURE

MMM define an object model for the management of application services and data from distributed and heterogenous provider sources. These objects define primary abstractions for all operation in the MMM infrastructure and constitute the basis for all user-system interaction, such as method and data entity, database loading, communication among system components, and communication with distributed application services.
All entities managed by MMM belong to one of the following classes: data set objects(DSO), method service objects(MSO), and method plan object(MPO). All objects instantiated through these classes support an extensible set of operation. All MMM object classes are derived from the abstract MMM Root Class which offers a common set of operations and attribute to its child classes.
The MMM middleware consists of the following key components: execution environment, service engine register, service engines, and MMM server. It implements several application-level protocols for inter-component communication.
The execution engine(EE) is responsible for the execution of methods. It schedules the method plans trying to utilize the available computing resources in an optimal manner. The EE incorporates several optimizations for minimizing data movement among computing resources and maximizing throughout. It knows about the available resources and performs load balancing. The EE receives a method plan object as input and returns a method plan object as output. The output MPO describes the result of the computation. It specified location of result data, computation time, possible error conditions, and the like. The execution environment interfaces with an application service through a service engine. It uses the service engine registry to obtain location details of the service engine(SE) driving the application service.
The service engine register(SER) is used to register and de-register application services and to maintain resources and availability information (location, network parameters, machine specification, etc.). The registry also maintains an up to date mapping of application service to service locations and resource characteristics. It thus provides vital information for the EE, which regularly queries SER for this information to decouple the EE from the service engines and allows for dynamic extensibility of the system. Application servers may be added at system run time without loss of operation and become immediately available as services. By frequently updating its resource list, the SER provides simple load balancing functionality that help to distribute the computational load across the available services, ensuring that no one service engine is overburdened.
The service engine(SE) encapsulate application services linking them into the MMM middleware. Service engine are responsible for converting data between the internal MMM format and the format required by the underlying application, retrieving data from remote sources, invoking the requested method on the application, and passing result information back to the MMM middleware. Note that “result information” can either be the result computation or a reference to the result of the computation, depending on the mode of invocation.
Each component in the MMM architecture is designed to be independent of implementation language and location within the distributed computing environment. Components can be configured to dynamically alter their information on other components. Components can be duplicated on the Web to increase availability.

4. MMM ARCHITECTURE FOR THE MICRO-OPTION

We know from the above that the MMM infrastructure is fit for the Micro-Option. Now, we give the gragh:



5. REFERENCES

[1] Roman Ginis and K.Mani Chandy:”Micor-Option:A Method for Optimal Selection and Atomic Reservation of Distributed Resources in a Free Market Evironment”
[2] H.-A. Jacobsen and O. Gunther:”Middle for Software Leasing over the Internet”
[3] Fetzer,C.and Cristian,F., “ An Optimal Clock Synchronization Algorithm,” Proceeding of the 10th Annual IEEE Conference on Computer Assurance,June 1995.
[4] Roman Ginis, “Opitimal Distributed Resource Allocation”,Masters Thesis, May 1999.California Institute of technology, tech report # CS-TR-99-08
[5] Lynch N., Merritt M., Weihl W., Fekete A., “Atomic Transactions,” Morgan Kaufmann Publishers,1994
[6] Beam,C c., Segev A., Shanthikumar J.G., “Optimal Design of Internet-based Auctions.” Working Paper 98-WP-1034, January 1999,University of California, Berkeley
[7] H.-A.Jacobsen and O.Günther. Component leasing on the World Wid Web. NETNOMICS.(accepted for publication)
[8] O.Günther., R. Müller, P. Schmidt, H. Bhargava, and R. Krishnan. MMM: A WWW-based approach for sharing statistical software modules. IEEE Internet Computing
[9] I. Foster and C. Kesselman. Globus: A metacomputering infrastructure toolkit. Int. Journal of Supercomputing Applications]

Survey of Weakly-Hard Real Time Schedule Theory and Its Application

Zhi WANG[1*] Ye_qiong SONG [2] Enrico-Maria POGGI [2] Youxian Sun[1]
(1 National Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China)
(2 LORIA–ENSEM, 2, av. de la Forêt de Haye, 54516 Vandoeuvre-lès-Nancy, France)

ABSTRACT

Normally, tasks are classified into real time and non real time according to temporal constraints for the processing and transmitting of these tasks, consequently the worst-case response time and average performance should be focused on them. However, in practical engineering context, partly violated temporal constraints can be tolerated if the violation meets certain distribution. Nevertheless, the loss-rate (within real time region, an instance of a task is regarded as loss if it violates its temporal constraint) under stable state or statistical real time can solve the problem in some extent, it can't include the permitted distribution of violation. For completely solving the problem, weakly-hard real time schedule theory or window-constraint real time schedule theory, which is used to investigate the problem related to allowing violation of instances over a finite range, consecutive instances or a time window, is proposed. In order to effectively utilize the fact that a practical application can tolerate some violations of temporal constraint under certain distribution, the fundamental research must be done from the aspects of specification of temporal constraint, schedule and schedulibility, and implementation, which are explained in detail in this paper.
Keyword: Weakly-Hard Real Time, (*m*, *k*)-firm, Real Time, Loss-rate, Quality of Service, Differentiated Service

1 INTRODUCTION[1]

Real time schedule theory is mainly applied to a kind of system, wherein the temporal aspects of their behavior are parts of their requirement. The correctness of the result of a task is not only related to its logic correctness, but also to when the result occurs. Normally, such system refers to real time system. Traditionally, within real time schedule theory, real time systems are classified into two types, HRT (hard real time) system and SRT (soft real time) system.

For applications with HRT requirements, non deadline missed is tolerated. It means that the time period from receiving a task to completing the task, refers to response time, must meet a temporal constraint enforced on this task, refers to deadline, otherwise the task comes to a failure. In practical engineering, many systems may be regarded as HRT system. For instance, within process control or manufacture, a controller has to collect message of controlled application from sensor. After processing the message, the controller transfers its command to an actuator, which directly controls application according to the command.

The traditional perspective of real time schedule theory has served the temporal safety-critical system community well. Most temporal safety-critical applications are regarded as HRT system on the temporal aspect of processing a task. To guarantee each instance of a task meet its deadline, a HRT system is designed and tested under the worst-case situation, where consider a task with its maximum executing time, and minimum arrival interval and minimum deadline. The analysis of such systems is performed with worst-case analysis to estimate an upper bound for application response time using service curve approaches [Cruz91] or classical worst-case response time analysis [Lehoczky90].

However there are still some pessimistic factors in them. Since even a safety-critical system, a computer automatic control system as an instance, not every task must be guaranteed its deadline because of the deadline being over looked. Within this kind of system, tasks periodically sample input signals, perform some computation and then send command to some actuators. Normally, the sample rate, which plays an important role in schedulibility, should be a multiple of the bandwidth frequency, the intrinsic physical characteristic of an application, therefore some deadline lost can be neglected if the lost do not occur consecutively over a long period. Besides, some signal interpolation techniques can compensate the deadline missed.

Normally, most no temporal safety-critical applications are regarded as SRT system. For SRT, it is permitted to miss some deadline occasionally. Examples of real time but non safety-critical application are multimedia applications, such as video-on-demand or streamed audio. It is important that information is received and processed at an almost constant rate, such as 30 frames per second for video information. However, some packets comprising of video frame can be lost, resulting in little or no noticeable degradation in the quality of service at the receiver. Similarly, a data source can lose certain fraction of information during its transfer across a network as long as the receiver can process the received data to compensate for the lost information. However，the term occasionally is not precise, but for SRT systems we should specify a probability to meet the deadline requirements. In general, the analysis of such systems is made using stochastic approaches and queuing theory [Takagi 90, King90, Rom90]

Therefore, there are several reasons why such a restricted perspective is proving increasingly inadequate for many emerging applications, which are not safety-critical, but nevertheless have significant real time performance requirement:

- A deadline missed in such system is usually not catastrophic; instead, such a failure typically leads to a gradual quality of service degradation.
- In practical engineering, the occasional loss of some deadlines usually can be tolerated due to over pessimistic assumption about temporal property of a task, such as its execution time, deadline arrival interval, and the robustness of related control algorithm to an application.
- Run-time scheduling algorithms are typically designed to be correct only on a feasible system (system in which is indeed possible to always meet all deadlines); on an infeasible system, such algorithm may perform unacceptably poorly. A good example of this is EDF algorithm, which is optimal under non-overload condition, but has been observed to perform miserably upon overload.
- Overload management should carefully discard instances

[1] This paper supported by NSFC-60203030, 60084001and PRA S101-04. Author of correspondence: zhiwang_iipc@yahoo.com or wangzhi@iipc.zju.edu.cn.

of tasks in order to improve effective utilization of resource and to minimize the amount of degradation caused by discarded instances.

Further, on the aspect of performance metrics, the worst-case performance is focused on in HRT system, but average performance is in SRT system. In fact, for a system, its behaviors are great difference between its worst-case situation and average performance. Design or select a system based on the worst-case performance is surely poor resource utilization. Not having to meet every deadline allows the capacity of real time system resource to be smaller than it would be to meet all of them. This permits the creation of simple and more cost-effective system that make better use of the available resources while guarantee, in the worst-case, a minimum level of services.

From the above analysis we know, most real time application can tolerate certain deadline lost. The situation that practical engineering requires for better characterizing the temporal constraints of real time tasks and effectively managing these tasks with these temporal constraints, make a new challenge to real time schedule theory. However, these two classes, HRT and SRT, might be insufficient to appropriately describe a real-time system. Traditional real time schedule theory only deals with how to guarantee deadline of each instance instead of allowing partly deadline missed. From this sense, traditional real time schedule theory greatly lags the requirement of practical engineering. Actually, for SRT systems, stochastic analysis gives only probability of deadline missed, and can not guarantee that these deadlines are missed in right manner to hold the good behaviour of real-time system.

Exploiting the emerging real time applications, which tolerate certain deadline missed provided that the deadline missed occurs in a clear, predictable and bounded way, has the following advantages:

- Alleviating the pessimism in parameter of system and worst-case scenarios.
- Providing a mechanism for fair degradation of quality of service
- Obtaining a fair mechanism for deciding which task need to be skipped during transient overload

To implement the above object, we need:

- Realize the behavior of a task in case some of its instances miss deadline
- Provide clear and intuitive constraints for specifying the number of deadlines missed and met over a period.

From the aspects of specification of temporal constraint, schedule and schedulibility (schedule analysis), and implementation, this paper gives introduction in detail.

2 SPECIFICATION OF WEAKLY HARD REAL TIME SCHEDULE THEORY

The above section suggests that most real time applications can tolerate certain deadline missed, but the corresponding real time schedule theory only investigates how to guarantee every deadline of a task under worst-case situation. Naturally, the lost rate, the percentage of deadlines to be met or missed (although it is common practice to do so), is the direct performance metrics for these real time applications, and based on which statistical real time channel with P[response time < deadline] > p is proposed[20]. However, this concept take deadline missed to be evenly distributed for granted. In fact it can't guarantee even distribution of deadline missed at all. For example, the requirement of a task like "less than 10% of deadlines can be missed" only represents average information over a large period of time. It may mean that one deadline is missed every 10 instances of the task or that 100 deadlines may be missed followed by 900 deadlines met. Clearly, the two cases are not the same. It appears that the tolerance to deadlines missed cannot be adequately specified by a single parameter. Up to present, only a little work is in the region of WHRT schedule theory. Although lots of works have been done to deal with real time problem, most work is focused on how to meet deadline of each instance in HRT context.

2.1 Weakly-Hard Real Time Schedule Theory

To deal with the practical issues, a new real time schedule theory is necessary. We refer this type of theory to weakly-hard-real time schedule theory. Formally,

Definition 1 WHRT (Weakly-Hard-Real Time) Schedule Theory: A weakly-hard-real time schedule theory is a conceptual framework which investigates the characteristics of a system that can tolerate certain deadline missed under a precise distribution over a finite time window. Hence, the tolerance to deadline missed is established within a window of consecutive invocations of the tasks.

Correspondingly, the temporal constraint under the context of WHRT schedule theory refers to weakly-hard-real time constraint (WHRTC).

2.2 Specification of WHRT Constraints[3,4,5,6,7][15,16]

To capture the situation that deadlines missed with a permitted distribution over a finite range can be tolerated, a second parameter describing the window of time within which the number of deadlines must hold should be specified. To address the problem, server QoS criteria have been proposed. First, Nagarajan proposes two criteria called interval QoS and Block QoS. Unlike a state-state QoS measure, the quality of service is measured over finite intervals of time. Nagarajan points state-state analysis is inadequate in minimizing the occurrence of high loss periods or in maximizing the occurrence of no loss period. However, Nagarajan has not provided any (m, k)-like WHRTC. In fact, the statistical real time channel proposed by King, essentially is a kind of state-state QoS although which provides a probability real time requirement over a point-to-point channel.

In this sense of WHRTC, the work is first done by Koren and Shasha, who propose an approach of description of deadline missed with deterministic distribution, skip factor. A task which has a skip factor of s will have one instance skipped out of its s consecutive instances. It is apparent that a skip factor at least deterministically guarantees at most one deadline missed occurring over a finite time, s consecutive instances. Further, Hamdauoi and Ramanthan expand the notion of the skip factor with (m, k)-firm, to specify a task that is desired to meet deadline of m instances among its consecutive k instances. Similarly, Richard and Christian propose windowed lost rate, that specifies a task can tolerate x deadline missed over a finite range or window, among consecutive y instances. Recently, Bernat and Burns summarize temporal properties of specifications available of WHRTC, point out the relations between various specifications. Further, they point out that a specification should be considered from two aspects: a task maybe is sensitive to the consecutiveness of deadline met while another is only sensitive to the number of deadline missed; a task maybe is sensitive to the consecutiveness of deadline missed while another is only sensitive to the number

of deadline missed. Concretely, they provide four types of basic specifications of WHRTC, these are:

- A task τ "meets any m in k deadlines", denoted with (m,k), if in any window of k consecutive instances of the task, there are at least m instances that meet the deadline.
- A task τ "meets consecutive m in k deadlines", denoted with $\langle m,k \rangle$, if in any window of k consecutive instances of the task, there are at least m consecutive instances that meet the deadline.
- A task τ " not misses any m in k deadlines", denoted with $(\overline{m},\overline{k})$ if in any window of k consecutive instances of the task, there are no more than m instances are missed
- A task τ "not misses consecutive m in k deadlines", denoted with $\langle \overline{m},\overline{k} \rangle$ if in any window of k consecutive instances of the task, it is never the case that m consecutive instances miss their deadline.

2.3 Relation between Different Specifications of WHRTC

For the present, although there are various specifications of WHRTC, fortunately, almost all of them have intrinsic relationships with the four types of basic specifications.

- Hard Real-time with response time<deadline is equivalent to $(k,\ k)$.
- Skip factor with skip one instance of s instances is equivalent to $(s-1,s)$ or $(\overline{1},\overline{s})$.
- Statistical real-time channel with P [response time < deadline] > p is equivalent to $p=\lim_{m,k\to\infty}\frac{m}{k}$.
- Windowed lost rate with tolerating x deadline miss over y consecutive instances is equivalent to $(y-x,\ y)$ or $(\overline{x},\overline{y})$.
- No real time just corresponds to (m, k) in the case of $m=0$.

Therefore, the four types of specifications are basis on the aspect of they can describe most specifications available.

Further, there are a certain relationship between these basic specifications, these are:

- $(m,k)=(\overline{k}-\overline{m},\overline{k})$
- $\langle \overline{m},\overline{k} \rangle = \langle \overline{m} \rangle$

2.4 Model of a Task with WHRTC

We only consider periodic tasks in the following, although tasks can be periodic or aperiodic (i.e. instances are randomly generated). In fact, in real-time community it is common to also consider sporadic traffic as periodic by taking the minimum inter-arrival time of instances as period. In practice, for most of transmission systems this minimum inter-arrival time does exist (e.g. 64-bytes packet + 96-bits IFS in Ethernet, leaky bucket smoothed input traffic).

We characterize a system with a set of n independent periodic tasks, $\Gamma=\{\tau_1,\tau_2,\cdots,\tau_n,\}$. Task τ_i can be described as the following model:

$$\tau_i=(O_i,\ D_i,\ T_i,\ C_i,\ \beta_i) \quad (1)$$

where O_i denotes release time of the instance job of τ_i, referred to initial time; D_i denotes maximum time allowed from the release time to the completion time of τ_i's instance, referred to deadline; T_i denotes interval between release times of two consecutive instances of τ_i, referred to period; C_i denotes maximum time needed to complete τ_i without any interruption, referred to execution time; β_i denotes WHRTC enforced to task τ_i. The j[th] instance of task τ_i is denoted as τ_{ij}.

Further, we can give definition of failure state and success state of task τ_i according to whether it meets its WHRTC.

Definition 2 Failure state and success state: a task τ_i is in success state if its last consecutive instances meet WHRTC, otherwise is in failure state.

From the above description, we can see a task may experience different states. Take (2, 3) of WHRTC as example. The state transition diagram of task model with WHRTC of (2, 3) is indicated in Fig.1.

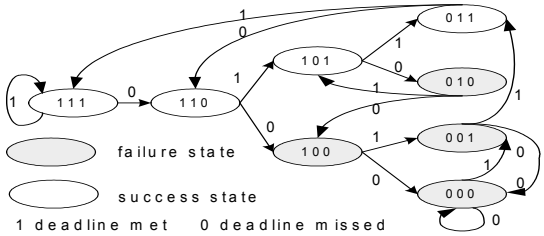


Fig.1 State transition diagram of task with WHRTC of (2, 3)

2.5 μ-pattern and WHRTC

The only information that a scheduler uses for tasks with WHRTC is the pattern of zeros and ones that represent missed and met deadlines on the history of the task (and possibly of the possible future of the tasks). These patterns refer to μ pattern[3][4].

Definition 3 μ-pattern: A μ-pattern of a task is a sequence of symbols of $\sum=\{0,1\}$ that characterizes the execution of the task. $|\mu|=p$ is the length of the pattern, and $\mu(k)\in\sum(1\le k\le p)$. 1 means that a task has met its deadline, and 0 means that the task has missed its deadline. $\mu(1)$ is the oldest invocation and $\mu(p)$ is the most recent invocation. An example with (4, 5)-firm constraint is given in Fig.2.

The μ-pattern is a word of k bits ordered from the most recent to the oldest invocation in which each bit keeps memory of whether the deadline is missed (bit = 0) or met (bit =1). In this paper, the leftmost bit represents the oldest. Each new invocation causes a shift of all the bits towards left, the leftmost exits from the word and is no longer considered, while the rightmost will be a 1 if the task has met its deadline (i.e. it has been served within) or a 0 otherwise.

In essential, all of these algorithms deal with a problem of partition of instances of a task to meet its WHRTC. Therefore, we can generalize the problem of scheduling tasks with WHRTC, further evaluate a schedule algorithm and investigate optimal schedule algorithm.

From definition 3 we can see that schedule for a task with a WHRTC is just to select a proper μ-pattern that meets the WHRTC. Therefore, whether there is an optimal μ-pattern among different μ-patterns that meets WHRTC is important.

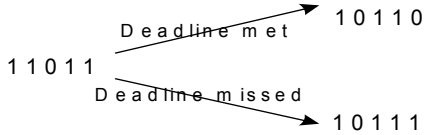


Fig.2 Possible evolution of the μ-pattern

Definition 4 Optimal μ-pattern : Given a task τ with a WHRTC, let the mandatory instances defined by a set of μ-pattern be assigned fixed priorities and the optional instances be assigned the lowest priority. The optimal μ-pattern is a μ-pattern that meets there are no other μ pattern can satisfy the WHRTC if the optimal μ-pattern cannot.

3. BASIC SCHEDULING ALGORITHMS OF WEAKLY HARD REAL TIME SCHEDULE THEORY

Although there are still lack general conceptual frame for weakly-hard real time schedule theory, in order to fairly distribute resource to meet temporal requirement of tasks with their respective WHRTC, there are parts of scheduling algorithms being proposed. The goals of these scheduling algorithms are various, part for providing deterministic guarantee of temporal requirement, part for improving flexibility by just providing best-effort service, part for implementing total performance when real time and no real time tasks co-exist. In this section, we just introduce some typical scheduling algorithms from different aspects under the context of weakly-hard real time schedule theory.

3.1 DBP (Distance Based Priority) Schedule[10]

For each τ_j , in order to guarantee its WHRTC, its priority is assigned based on the number of deadline misses that task can still stand before violating its (m, k) requirement. This allowing number of deadline misses is referred to as distance, i.e. the distance to a failure state from current state. When a task is violating is (m, k) requirement, that task is said to be in failure state. The evaluation of this distance can be done exactly considering the recent history of τ_j .

Fig.1 suggests that the closer of a task to its failure state, the more easily the task suffers failure. That activates the idea of DBP schedule. As for an instance of a task with WHRTC of (m, k), DBP designs priority to the instance according to the information of its last k consecutive historical instances. Further, Fig.1 also suggests that a task with WHRTC of (m, k), the trend of its state to failure state is relevant to the position of m^{th} 1 (position of m^{th} deadline meet occurs) from the last k instances.

The priority assigned by DBP to a job at a given instant is equal to the distance of the current μ-pattern to a failure state. This distance can be easily evaluated, by adding in the right side 0s until failure state and the number of added 0s is the priority. If a stream is already in failure state (i.e., less than m 1s in the μ-pattern), the highest priority 0 is assigned. This is also given by equation (1). For example, considering a task with (3,5)-firm constraint, the current job j_{i+1} is set the priority of 2 if its previous 5 consecutive jobs construct the state of (11011), and is set the priority of 3 if its previous 5 consecutive jobs construct the state of (10111). Formally,

Let $s_j = \left(\delta^j_{i-k_j+1}, \cdots, \delta^j_{i-1}, \delta^j_i\right)$ denote the state of the previous k consecutive instances of task τ_j, $l_i(n,s)$ denote the position of n^{th} meet in the state of s_i, then the priority of current instances of task τ_i is

$$priority_j^{i+1} = \begin{cases} k_j - l_j(k_j, s_j) + 1 & \left(l_j(k_j, s_j) \le m_j\right) \\ 0 & \left(l_j(k_j, s_j) > m_j\right) \end{cases} \quad (2)$$

For example, considering a task with (3, 5) of WHRTC, the current instance of the task is set the priority of 2 if its previous 5 consecutive instances construct the state of (11011), and is set the priority of 3 if its previous 5 consecutive instances construct the state of (10111).

One of the problems faced with DBP, is that it assigns priorities only considering one τ_j without comparing it to the others sharing the same server. This self-reference behaviour may lead to a situation where more than one stream get the same priority at the same time, in this case an algorithm to choose among them should be defined. It is also important to underline that DBP chooses priority based on the history of the stream's μ-pattern, and doesn't take into account any specific information on the actual attributes of the stream like its length c_j, its minimum inter-arrival time T_j , and its deadline D_j.

The simplest and common way to overcome these problems is to assign DBP-based priority to the jobs and, in case of priority equality, use another scheduling algorithm among the already known ones. In their paper, Hamdaoui and Ramanathan [Hamdaoui95] combined DBP with Earlier Deadline First (EDF). However this solution gives to Deadline less importance than that given to the μ-pattern, since EDF would be used only when μ-pattern is not sufficient, i.e. when two streams get the same DBP-priority.

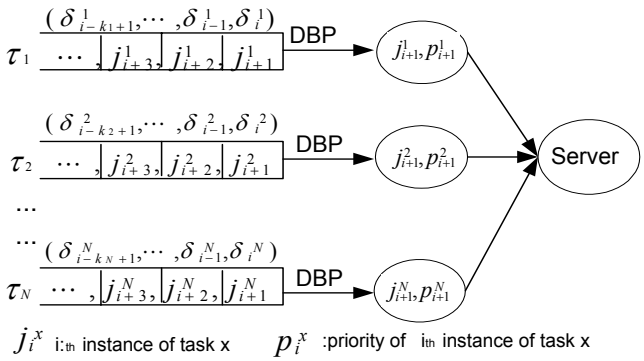


Fig.3 The Implementation of DBP Schedule Algorithm

To overcome the limits of DBP, the properties of periodic tasks, and the relationships among these properties should be considered, and an improved DBP, which integrated the above neglected factor should be researched in detail.

3.2 DWC (Dynamic Window Constrained) Schedule[16]

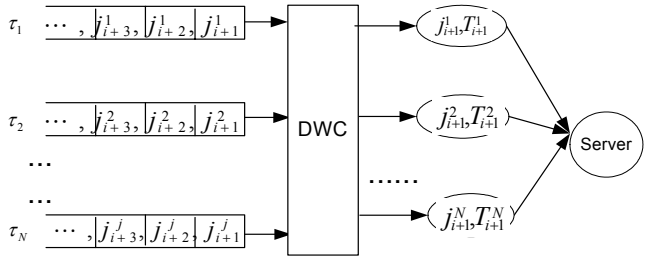
DWC schedule maintains information of each task just like DBP schedule does, but how to utilize the information is significantly different from DBP schedule. Whereas DBP schedule processes the current instance a task using its the state transition and its k last historical instances to capture the relative priority of a task, DWC schedule using the notion of dynamic window in which m and k are allowed to change. In DWC schedule, each time an instance of a task τ_i is transmitted or dropped, the information of (m_i, k_i) is adjusted accordingly.

Table 1 Main rules of DWC for Assigning Priority to an Instance

Lowest loss-tolerance first
Same non-zero loss-tolerance, order EDF first
Same non-zero loss-tolerance & deadlines, order lowest loss-numerator first
Zero loss-tolerance & denominators, order EDF first
Zero loss-tolerance, order highest loss-denominators
All other cases: first-come-first-serve

The main rules for DWC schedule are as follows: DWC schedule algorithm processes instances of tasks based on the current values of their loss-tolerance and deadlines, and gives

precedence to the instance with the lowest loss-tolerance. Instances of the same task all have the same original and current loss-tolerance, and are scheduled in their order of arrival. The loss-tolerance of an instance (and hence the corresponding task) changes over time depending on whether or not another earlier instance from the same task has been scheduled for transmission by its deadline. Whenever an instance misses its deadline, the loss-tolerance for all ongoing instances in the same task is adjusted to reflect the increased importance of transmitting these instances. This approach avoids starving the service granted to a given task and attempts to increase the importance of serving any instance in the task which is likely to violate its original loss constraints. Conversely, any instance serviced before its deadline causes the loss-tolerance of other instances (yet to be serviced) in the same task to be decreased, thereby reducing their priority.



j_i^x:Ith instance of task x T_{i+1}^x :loss_tolerance of i th instance of task x

Fig.4 The Implementation of DWC Schedule Algorithm

3.3 ERM (Enhanced Rate Monotonic) Schedule[11]

DBP and DWC schedules essentially are dynamic priority based schedule and belong to best-effort, that means they can't provide any deadline guarantee at all. At least, there is still not an effective approach to check whether they guarantee a task meet its WHRTC of (m, k). In fact, the goal of any schedule is to effectively manage tasks and distribute proper resource for these tasks. Because most static priority based schedule algorithms can provide deterministic guarantee of a task temporal constraint, and because the essential of WHRTC of (m, k) is to meet m of k deadlines, it is possible to construct a static priority promotion approach that can select m instances from any consecutive k instances. If we can guarantee the selected m instances of a task from any its k consecutive instances, we can at least meet WHRTC of (m, k) of this task. From this sense, the idea of Imprecise Computation (IC), that divides instances of a task into a mandatory and an optional part and the latter is rejected when system overload, is the same as WHRTC of (m, k). Further, Rate Monotonic (RM) schedule, a well-known static priority schedule algorithm for periodic tasks, effectively schedule periodic tasks based on the tasks' period. Combining the ideas from the IC and RM schedule, other schedule, ERM schedule, is proposed. That is the instances of a task is divided into a mandatory and an optional part, and the instances of mandatory part are scheduled according to Rate Monotonic policy, and instances of the optional part are assign the lowest priority and scheduled according to First Come First Service (FCFS).

The implementation of ERM schedule is very simple and easy; the key problem is how to select mandatory part of instances for a task. An approach for classification of instances of task τ_i as mandatory or optional is given, that is based on the value of m_i and k_i.

The instances of task τ_i activated at $(a = 0,1,\cdots,)$ are classified as mandatory if a meets

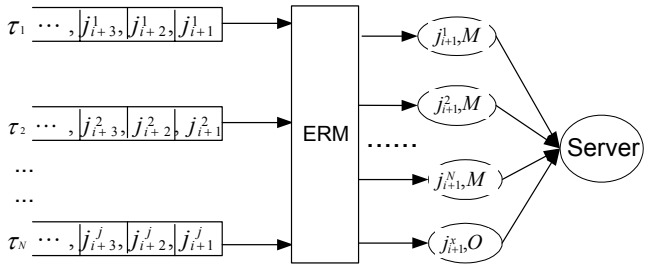
$$a = \left\| \left\lceil \frac{a \cdot m_i}{k_i} \right\rceil \cdot \frac{k_i}{m_i} \right| \quad (3)$$

Take the following two tasks as example:

$\tau_1 : C_1 = 1,\ T_1 = 2,\ m_1 = 2,\ k_1 = 3,$

$\tau_2 : C_2 = 3,\ T_2 = 4,\ m_2 = 2,\ k_2 = 3$

For task τ_1, one out of its every three instances is classified as optional, starting with the instance activated at time 8. For task τ_2, the instances with activation times 24, 48, 84, 108, … , are classified as optional.



j_i^x:Ith instance of task x M :mandatory task O :optional task

Fig.5 The Implementation of ERM Schedule Algorithm

Take ERM schedule as example, it has the following property from the aspect of selecting a proper μ-pattern,

- ERM implicitly selects mandatory instances of a task evenly among its m consecutive instances, and solely depends on the ratio of m over k of the task.
- The first instance of every task is always designated as mandatory or given the highest priority.

Apparently, judicious selection of mandatory v.s. optional instances and promotion of priority for instances play a critical role in scheduling tasks with WHRTC. However, ERM has following intrinsic disadvantages:

- Regardless of period and execution time, the mandatory instances of two tasks, having the same ratio of m over k, are always distributed in the same way among the m consecutive instances.
- Such even distribution lacks flexibility and may not be advantageous in certain situations.

4. EXPEBDED SCHEULING ALGORITHMS OF WEAKLY HARD REAL TIME THEORY

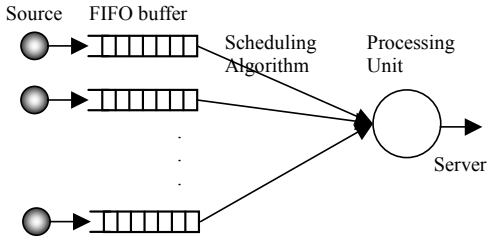


Fig.6 MIQSS model

In fact, all the above scheduling algorithms are basic scheduling algorithm in the sense that they are applied in MIQSS (multiple input queues single server) and under the condition where that only real time tasks with WHRTC exist. MIQSS model can be used to study a large category of computer and telecommunication systems such as multiple tasks execution in a CPU, transmission of messages issued from multiple message sources sharing a same transmission medium or network interconnection equipment. The proposed model is made up of *N* sources generating *N* streams of jobs τ_i (*i =1, 2, ... N*) attempting to be served by a single server. Each

stream is formed by a source and a waiting queue, where a job (can represent a task or a message) issued from the source waits until chosen by the server. The server chooses jobs at the head of queues according to its scheduling policy.

However, in most actual applications, tasks or messages are diversity on the aspect of hops of end-to-end connection over which tasks have to transfer, the number of tasks and types of tasks a server processes.

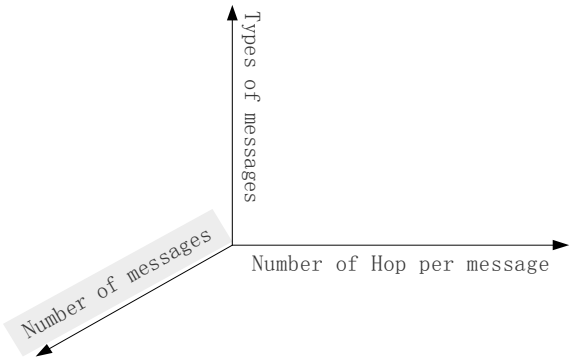


Fig.7 Complexity of schedule algorithms within WHRT

- **Multi-hop end-to-end connection**

Within a distributed real time system, a task needs data packet crossing multiple networks, multi-hop connection, and accordingly requires QoS in end-to-end. However, on the aspect of multi-hop, most schedule algorithms, such as DBP, DWC and DCQ, are not applicable any more. It is because the deadline in DBP like schedule algorithms is only a local deadline in multi-hop. Correspondingly, the rejected packet due to exceed its deadline is actually exceeding its local deadline in multi-hop, and maybe still have a chance to meet its end-to-end deadline.

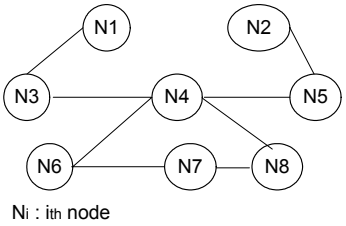


Fig.8 Network Topology of a Multi-hop Network

Parameter of Task

Task	Route	Deadline	(m,k)
T1	N1-->N3-->N4-->N8	30ms	(3,4)
T2	N2-->N5-->N4-->N6	25ms	(5,7)
T3	N3-->N4-->N5	15ms	(5,6)
T4	N6-->N7-->N8	30ms	(3,4)
T5	N7-->N8-->N4-->N3	24ms	(6,7)
T6	N8-->N4-->N5-->N2	35ms	(1,1)

Fig.9 Parameter in a Multi-hop Network

For example, suppose a network of multi-hop in Fig.8,9, which has 8 nodes $n_i \ (1 \le i \le 8)$, and 6 tasks $\tau_i \ (1 \le i \le 6)$. These nodes are connected by point-to-point, data packets of tasks are transmitted across multi-hop connected by nodes. For task τ_2, which must travel three hops of $n_2 \to n_5 \to n_4 \to n_6$ and has an end-to-end deadline of $D_i = 25ms$ and WHRTC of $(5, 7)$. That means an data packet of τ_2 generated from its source node of n_2 at time t must reach its destination node of n_6 at time $t + D_i$, and its at least 5 data packets meet end-to-end deadline among its every 7 consecutive data packets. For the present, in the context of multi-hop end-to-end connection, only EDP-M (Modified-DBP) and EDBP (Enhanced EDP-M) are proposed and implemented.

- **Number of Tasks**

All of the schedule algorithms of DBP, ERM and DWC are common in this aspect, the schedulers utilize information of each task. Correspondingly, the above schedule algorithms are tasks-aware. However, the overheads of maintaining the information will rapidly increase with the increasing number of tasks. Example of application is real time media server, which will be responsible for lots of client, with a wide range of QoS requirement.

To overcome the limitation, the schedule algorithms must be scalable, where scalable means the overheads of schedule algorithms is independent to the number tasks in these schedule algorithms. Scalability has get attention in IETF, which focuses on providing architecture for real time service through Internet. Two systematic approaches have been proposed by IETF, one is InteS (Integrated Service), another is DiffS (Differentiated Service). InteS provides real time service by processing information of each task, and accordingly does scale well. Contradictorily, DiffS provides real time service by class-based schedule policy, wherein each task is partitioned into correspondingly class. In fact, the information of each class is the result of aggregation of tasks belonged to this class. On the aspect of scalability, per class based schedule algorithms scale well than per task based, but there are fundamental problems to be solved on task with WHRTC. One of them is partition of tasks with WHRTC into a class, which is concerned with relationship of various WHRTC, such as whether a WHRTC being hard than another on the aspect of temporal constraints, how to determine a general metrics to various WHRTC. Up to present, only DCQ (Dynamic Class-based Queue) is proposed.

- **Types of Tasks**

Because of diversity of tasks on the aspect temporal requirement in an application or a system, not all tasks are real time, and part tasks maybe no real time. Therefore, effectively scheduling SRT task while guarantee the behavior of HRT task is becoming problem, and many techniques have been proposed to solve the problem. In fact, in many real time systems, hard real time task and soft real time task co-exist. Within this situation, a number of approaches have been proposed to deal with this mixed task set, such as DS (Deferrable Server), PE (Priority Exchange), SS (Slack Stealing), and DP (dual priority). Among these techniques, dual priority schedule is an intuitively simple method and lower overhead.

Although DP has some advantages in its simplicity and low overhead compared to other approaches, nevertheless B. Ganja has proved that DP is not always better than background scheduling on the aspect of improving responding time of no real time task[25]. For the present, to my best knowledge, EDP (Enhanced Dual Priority) is the only schedule algorithm being investigated under the co-existence of tasks with WHRTC and no real time tasks, let alone comparison among these schedules.

4.1 BDP-M and EDBP: Schedule Algorithm under Multi-hop End-to-end Connection

It is obvious that the DBP like schedule algorithms can't be directly used because all of them are applied under the assumption that sever can determine whether a task misses or meets its deadline. However, when a task must be relayed through multiple nodes, then all intermediate nodes can't locally determine whether the task meet its end-to-end

deadline. Apparently, a direct approach is distributing end-to-end deadline of a task into local deadline of hops where the task has to cross.

The simplest approach is evenly distributing end-to-end deadline according to the number of hops. In fact, BDP-M (Modified-DBP) and EDBP (Enhanced EDP-M) belong to the simplest approach on the aspect of distributing end-to-end deadline. The main idea of these two approaches is first check whether an instance (data packet) of a task violate its end-to-end deadline; then the selected instance is scheduled according EDP schedule.

- **Adaptability of Schedule Algorithms under Multi-hop End-to-end Connection**

On the aspect of multi-hop, adaptability of schedule algorithms is concerned with how distributing WHRTC of end-to-end (referred to global WHRTC here) into local WHRTC, and adjusting local WHRTC online to adapt actual situation. Solving the above problem consists of two challenges.

The global policy for dealing with data packet transmission across multi-hop:

- How to design end-to-end deadline of a task along the hops over which the task transfers
- How to design WHRTC of (m, k) of a task along the hops over which the task transfers

The local policy for dealing with data packet transmission in local hop:

- How to processes a task to adapt the actual requirement in its global WHRTC. A local deadline has two types, actual delay and setting deadline (given by global policy), and local policy processes instance of a task according to its local WHRTC, including local deadline and local (m, k). Presently, EDP-M and EDBP only locally process its instances of tasks separately, without considering the actual global situation of these tasks, although the both schedule algorithm do check of whether an instance exceeding its end-to-end deadline first. In fact, if the actual delay of a task in a local hop is near to end-to-end deadline, the task will get more chance to be processed if its local deadline decreased correspondingly. The reason is simple, the shorter local deadline, the more probability of deadline violation, and accordingly higher priority. Intuitively, it is a complicated problem, but we can get some general guidance through simulation.

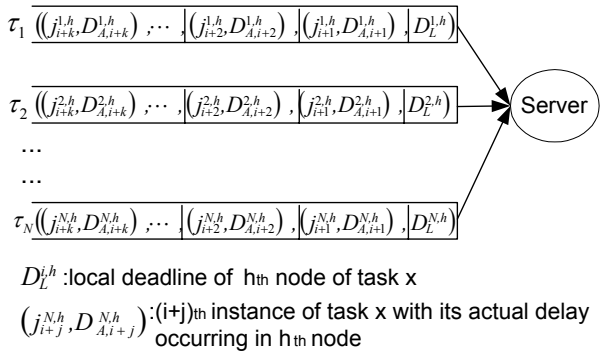


Fig.10 The Implementation of EDBP Schedule Algorithm

4.2 DCQ: Schedule Algorithm Being Scalable

As for QoS of Internet and scalable, DCQ schedule algorithms is first proposed to deal with tasks with WHRTC. The key issue to DCQ schedule has to solve is group membership, that consists of mapping a task with WHRTC into a class, and automatically adapting membership of each class when a task joining or leaving. DCQ schedule implements its goal through two level schedules, the first one is dealing with group membership, the other is scheduling each class according a given schedule algorithms, such as DBP schedule and DWC schedule.

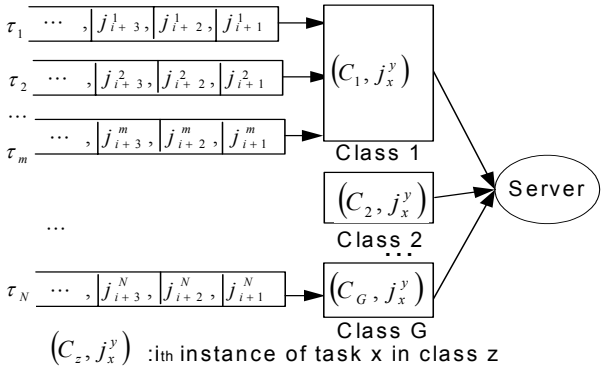


Fig.11 The Implementation of DCQ Schedule Algorithm

4.3 EDP: Schedule Algorithm for Co-existence of Tasks with WHRTC and SRT Tasks

DP schedule consists of three priority bands, there are lower, middle and upper. A HRT task is assigned two priorities, lower and upper, and the SRT task is only assigned middle priority. Upon invocation of an instance of a HRT task, the instance is assigned a low priority and it is promoted to a high priority to guarantee the deadline of the HRT task be met.

The key problem for dual priority schedule is to select proper promotion time Y_j for an instance of task τ_j after the instance is released.

$Y_j = D_j - R_j$

(4) where, R_j is response time of task τ_j.

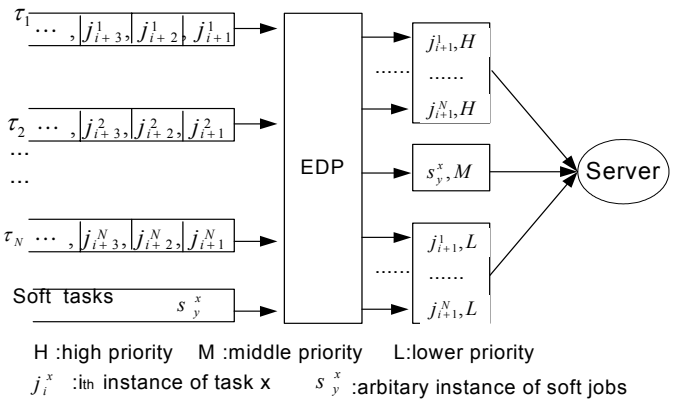


Fig.12 The Implementation of EDP Schedule Algorithm

The later a task has its priority promoted, the more slack time is available for SRT tasks and the better it is distributed. Consequently response time of SRT task is improved. Unfortunately, the value of Y_j may be quite close to the invocation if there is tight deadline for task τ_j. If we exploit the fact that tasks with WHRTC can tolerate some of their deadline missed, we can improve the responsiveness of SRT tasks. The improved DP refers to EDP Schedule.

This can be done by the following two approaches:

- Optimizing promotion time Y_j, so that tasks meet their WHRTC of (m, k).

- Optimizing selection strategies, thus offering more computation time to middle band for soft tasks whilst still guaranteeing real time tasks with WHRTC of (m, k).

5. PROBLEM YET TO BE SOLVED

Although many works have been done in this region, there are still some basic issues to be solved.

5.1 Specifications of WHRTC

Although Bernat and Burns propose four types of basic WHRTC, which are still lack enough capability for describing temporal constraint of a task. Take $\langle \overline{m}, \overline{k} \rangle$ as example, a μ-pattern of $(0,0,0,1,0,0,0,1,0,0,0,1,\cdots,)$ seems un-convincible despite of it meeting the requirement of $\langle \overline{3}, \overline{4} \rangle$. For solving this problem, $\langle \overline{m}, k \rangle$ like WHRTC maybe more convincible, where $\langle \overline{m}, k \rangle$ denote a task τ "not misses deadlines of m consecutive instances and at least meets deadlines of k consecutive instances". Similarly, there are other different WHRTC. It is note that the WHRTC should be easy implemented and analyzed.

5.2 Dynamic and Static Schedule Algorithms

DBP and DWC are dynamic schedule, on the aspect that priority of each instance of a task is automatically adjusted according the information of the previous instances of the task. For the present, almost all are dynamic schedule algorithms are DBP or DWC like, there is lack new idea on how to adjust priority of each instance dynamically.

ERM is static schedule, on the aspect that priority of each instance of a task is fixed, which only depended on the ratio of m to k in its WHRTC, instead the previous instances of the task. For the present, only one static schedule, ERM, is proposed and investigated.

Further, all schedule algorithms available focus only on WHRTC of (m,k), no matter static schedule or dynamic schedule, there are still lack all schedule algorithms on other WHRTC, such as $\langle m, k \rangle$, $(\overline{m}, \overline{k})$ and $\langle \overline{m}, \overline{k} \rangle$.

DBP and DWC are dynamic schedule, on the aspect that priority of each instance of a task is automatically adjusted according the information of the previous instances of the task. For the present, almost all are dynamic schedule algorithms are DBP or DWC like, there is lack new idea on how to adjust priority of each instance dynamically.

ERM is static schedule, on the aspect that priority of each instance of a task is fixed, which only depended on the ratio of m to k in its WHRTC, instead the previous instances of the task. For the present, only one static schedule, ERM, is proposed and investigated.
Further, all schedule algorithms available focus only on WHRTC of (m,k), no matter static schedule or dynamic schedule, there are still lack all schedule algorithms on other WHRTC, such as $\langle m, k \rangle$, $(\overline{m}, \overline{k})$ and $\langle \overline{m}, \overline{k} \rangle$.

5.3 Schedulability and Optimization of Schedule Algorithms

Providing online or offline schedulability checks for determining whether tasks with WHRTC being satisfied under given load and schedule algorithm is critical.

Schedulability checks is discovered as checks μ-patterns, nevertheless solving the problem consists of two challenges:

- How to determine if one set of μ-patterns is better or easier to be scheduled than another under a given task set with a WHRTC?
- How to predict if the corresponding mandatory jobs are all schedulable under a given set of μ-patterns?

Unfortunately,

- Searching optimal μ-pattern for each task is a NP-hard problem.
- Determine the schedulability of arbitrary μ-pattern of a task with a WHRTC is NP-hard problem.

It is well known that there are two main approaches for schedulability of a schedule algorithm. However, it is fronted great change in the calculation of utilization of resource and worst-case responding time (WSRT). Take the second as explanation; the key of calculation of WSRT of a task is in calculation of its busy period. However, the busy period of a task with WHRTC fronts the following challenges:

- Critical instant, the instant that all tasks arriving with their maximum is not critical instant in the sense of meeting WHRTC of a task, which is related to the information of previous instance. It is obvious the critical instant in traditional analysis is not the instant that a task is nearest to its failure state.
- Feasible load at time t, in traditional analysis and load at time t is very easy, but the situation is serious because for a task with WHRTC, wherein the distribution of discarded instance must be considered.

Intuitively, determine the WSRT of a task with WHRTC is NP-hard problem.

The schedulability of a task with a WHRTC and its sub-optimal μ-pattern can be further improved if one can tolerate spending more time on finding better mandatory/ optional partitions off-line. In this regards, a probabilistic optimization algorithm can be very effective[8][9].

- Genetic algorithms
- Simulated annealing

One challenge in applying such schedule algorithms is to formulate an appropriate objective function.

5.4 Sensitiveness of Tasks with WHRTC

For a system, we are interesting to schedulable or un-schedulable of the system, but also interesting to the margin by which the system un-schedulable or schedulable. On one aspect, how much reduction of capability of the system, such as rate of CPU and bandwidth of network, is allowed, or how many tasks are allowed to join if a system is schedulable. On another aspect, how much addition of capability of the system is needed, or how much WHRTC of tasks is needed if a system is un-schedulable. This type of analysis refers to sensitiveness analysis (SA). In fact, for task with WHRTC also needs SA. SA is important to admission control of network, especially in the negotiation between user and provider of network service. Traffic Model of Task with WHRTC

5.5 Traffic Model of Task with WHRTC

For the present, most researches focus on periodic task, except for a few on aperiodic tasks with Poisson arrival. Actually, within Internet most tasks is described by a traffic model (σ, ρ) proposed by Cruz, instead of by either periodic task or by aperiodic task. Within (σ, ρ), σ denotes arrival average rate of task, and ρ denotes burst size. It is feasible

that we research tasks with WHRTC under the frame that the tasks have temporal characteristic of (σ, ρ) arrival.

6. CONCLUSION

With the emergence of lots of real time applications that can tolerate a certain deadline missed, the understanding real time must be improved instead of just guaranteeing deadline of each instance of a real time task. Accordingly, the real time schedule theory need be expanded to investigate the new phenomena. This paper summarizes the-state-of-art of weakly real time schedule theory on the aspect of specification, schedule algorithm and schedulability, and its applications. Further, the basic issues to yet be solved are pointed out in this paper.

7. REFERENCES

[1] G. Bernat and A. Burns. Combining (n, m)-hard deadlines and dual priority scheduling. *Proceedings of Real-Time Systems Symposium*, pages 46–57, Dec 1997.

[2] G. Bernat and A. Burns. Weakly_Hard real-time systems, *IEEE Transactions on Computer*s, 50(4),pp.308-321,2001.

[3] G. Bernat and R.Cayssials, Guaranteed On-line Weakly_Hard real-time systems, *Proceedings of IEEE conference on real-time system*s, 2001.12

[4] G. Quan and X. Hu. Enhanced Fixed-priority Scheduling with (m, k)-firm Guarantee, *12th IEEE Europmicro Conference on Real-Time System,* 2000

[5] J.-Y. Chung, J. W. Liu, and K.-J. Lin. Scheduling periodic jobs that allows imprecise results. *IEEE Transactions on Computer*s, 39(9):1156–1175, Sep 1990.

[6] M. K. Gardner and J. W.S.Liu. Performance of algorithms for scheduling real-time systems with overrun and overload. *Proceedings of the eleventh euromicro conference on real-time system*s, pages 287–296, Jun 1999.

[7] K. Gilad and S. Dennis. Dover: an optimal on-line scheduling algorithm for overloaded uni-processor real-time systems. *Electronics Letter*s, 33(15):1301–1302, July 1997.

[8] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learnin*g. Addison-Wesley, MA, 1989.

[9] F. Remeo, *Simulated Annealing: Theory and Applications to Layout Problem*s. PhD thesis, Dept. Of Elec. Eng.& Comp. Sci., University of California, Berkeley, Mar. 1989.

[10] M. Hamdaoui and P. Ramanathan. A dynamic priority assignment technique for streams with (m, k)-firm deadlines. *IEEE Transactions on Computer*s, 44(4), 1443–1451, Dec.1995.

[11] M. Hamdaoui and P. Ramanathan. Evaluating Dynamic Failure Probability for Streams with (m, k)-firm Deadline. *IEEE Transactions on Computer*s, 46(12), pp.1325–1337, Dec.1997.

[12] W. lindsay and P. Ramanathan, DBP-M, A Technique for Meeting end-to-end (m, k)-firm Guarantee requirements in point-to-Point networks, *Procs of IEEE Conference on Local networks,* pp.294-303, 1999.

[13] P. Ramanathan. Overload management in Real-Time control applications using (m, k)-firm guarantee. *IEEE Transactions on Parallel and Distributed System*s, 10(6):549–559, Jun 1999.

[14] C.C. Han and H.-Y. Tyan. A better polynomial-time schedulability test for Real-Time fixed-priority scheduling algorithms. *Proceedings of the Real-Time Systems Symposium*, pp36–45, 1997.

[15] G. Koren and D. Shasha. Skip-over: Algorithms and complexity for overloaded system that allows skips. *Proceedings of Real-Time Systems Symposiu*m, pages 110–117, Dec. 1995.

[16] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *Proceedings of 1989 IEEE Real-Time System Symposiu*m, pages 166–171, 1989.

[17] A. Striegel and G. Manimaran, Dynamic Class-Based Queue Management for Scalable Media Servers, to appear in *Journal of Systems & Software*

[18] A. Striegel, G. Manimaran, Packet Scheduling with Delay and Loss Differentiation, *Computer Communications,* vol.25, no.1, pp.21-31, Jan. 2002.

[19] A. Striegel, G. Manimaran, Best-effort Scheduling of (m,k)-firm Real-time Streams in Multihop Networks, *Workshop of Parallel and Distributed Real-Time Systems*, Mexico, 2000.

[20] 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

[21] R. West and C. Poellabauer, Analysis of a Window-Constrained Scheduler for Real-Time and Best-Effort Packet Streams, *Proc. of 21st IEEE Real-Time Systems*

[22] R. West, K. Schwan, and C. Poellabauer, Scalable Scheduling Support for Loss and Delay Constrained Media Streams, *Proc. of IEEE Real-Time Technology and Applications Symposium,* 1999

[23] B. Choi, D. Xuan, C. Li, R. Bettati, and Wei Zhao, Scalable QoS Guaranteed Communication Services for Real-Time Applications*, Proc. of the IEEE International Conf. on Distributed Computing Systems,* April 2000.

[24] S. Wang, D. Xuan, R. Bettati, and Wei Zhao, Providing Absolute Differentiated Services with Statistical Guarantees in Static Priority Scheduling Networks, *Prods of IEEE Real-Time Technology and Applications Symposium (RTAS)*, 2001.

[25] B. Gaujal, N. Navet and J. Migge, Dual-Priority versus Background Scheduling: A Path-Wise Comparison, to appear in Journal of Real Time System

[26] R. L. Cruz, Quality of Service Guarantees in Virtual Circuit Switched Networks, *IEEE Journal of Selected Areas in Communication*, 1995.

[27] R. L. Cruz, A Calculus for Network Delay, *IEEE Transactions on Information Theory* (1, 2), January 1991.

[28] P.J.B King, Computer Communication Systems Performance Modelling. Prentice Hall, 1990

[29] C. Buttazzo and Giorgio, Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Application, Kluwer Academic Publisher, 1997

[30] H. Takagi, Queuing Analysis of Polling System: An Update in Stochastic Analysis of Computer and Communication System, Elsevier Science North Holland, Amsterdam, 267-318, 1990

[31] R. Rom, M. Sidi, Multiple Access Protocols_ Performance and Analysis, Springer Verlag new York, 1990

A Concurrency Control Mechanism — Traffic Light Model and its Application in Database System

Qingsheng Zhu, Donggen Guan, Weiwei Li
Dept. of Computer Science, Chongqing University, 400044, China
E-mail:qszhu@cqu.edu.cn

ABSTRACT

In this paper we design a traffic light model to complete concurrency control in database systems. This model can process the priority of reading operation and writing operation. The elementary unit processed in our model is not the transaction but the operation, thus it can be more convenient and more precise to control users' operations in a database system. The paper firstly introduces the important significance of concurrency control in database system and describes the relation between concurrency control and data consistency. Then it analyses the principle of traditional concurrency control method and 3-level lock protocols. Finally it proposes a new concurrency control mechanism — traffic light model for application of database system.

Keywords: Concurrency Control, Data Consistency, Traffic Light Model

1. INTRODUCTION

Concurrency control is an issue that we generally pay attention to in distributed system, multi-process system and multi-thread system. Its main goal is to ensure to gain expected result when the shared resource was accessed concurrently. For example, it is necessary to lock the operated objects in computer supporting cooperative writes in order to ensure data consistency. Database is one kind of shared resource and can be used by many users concurrently. When these users' programs serially operate on database one by one, only one program can be executed at each moment. One-program accesses the database and other programs have to wait until it has been completed. But if a user's program involves a great deal of I/O exchange, the database system will be idle in much of time. Therefore, in order to make use of database resource as full as possible, more then one users should be permitted to access database in parallel. However it will bring the problem that several users' programs maybe access the same data concurrently. If the concurrency operation cannot be controlled in effect, it may result in accessing or saving incorrect data, which may destroy database consistency. For this reason, the database management system must offer concurrency control mechanism. The concurrency control mechanism is an important mark in scaling the performance of database management system. In a database system, a group of operations on database are generally managed as some transactions with ACID (atomicity, consistency, isolation and durability) properties. The scheduling system views the transaction as basic unit and uses lock or time stamp to ensure correctness in scheduled transactions. Although these methods have obtained good effect and efficiency in centralizing structure, they have some disadvantages. For example, if concurrency operations are not controlled properly, they may result in inconsistency of data in database. The inconsistency includes lost update, non-repeatable read and dirty-data read.

Lost Update means that transaction A and transaction B read and modify the same data from a database so that the action of transaction B destroys the result of transaction A, and lead to lose the update of transaction A. This process is shown as Figure 1(a).

Non-repeatable Read denotes that transaction B executes update-operation after transaction A read data, and then transaction A cannot get the last reading result, as shown in Figure 1(b).

Dirty-data Read indicates that transaction A modifies data and writes it back to disk, and then after transaction B read the same data, the transaction A is canceled for some reasons. At that time transaction A is rollbacked, and the data which transaction B read is not consistent with that in database. The data is incorrect, and it is called dirty-data, as shown in Figure 1(c).

2. TRADITIONAL CONCURRENCY CONTROL

An important characteristic of traditional methods is to make use of some concurrency control strategies (e.g. serializability, lock, time stamp and so on), which applied often in Database System or Operation System to maintain the consistency of copy object. It will be found out that these methods have some disadvantages because of the difference of computing environment,. The method of lock is actually that the update and access permission to shared objects is granted to a particular visitor in order to make the shared object accessing serializable for ensuring data consistency. In this strategy, there must be a centralized scheduling mechanism or a distributed scheduling mechanism to finish the assignment. Aiming at some different purposes in accessing shared object, lock can be divided into different classifications. When some site requests some kind of lock for some shared object, scheduling mechanism will determine whether to approve new lock request according to the compatibility regular between the lock which has been locked on requested object and the lock which is now requested. Any site can access correspond objects by means of the method which is applied only after satisfaction of lock request. And it should release the lock as soon as the access ends in order to satisfy with the lock request of other sites.
Based on whether the process of operation need to pause before the lock is permitted and whether the tentative lock can be released before permitted, the lock strategies can be divided into two categories: pessimistic lock and optimistic lock.

Pessimistic Lock Strategy: If you use this strategy you cannot process the operated object until you get the permission of corresponding lock. In the 'jam mode', the user interface will stop the response to user's operation until receiving the response to the lock request. The operated object will be transacted as soon as the permission is gained. In the 'non-jam mode', the system can deal with some other operations while

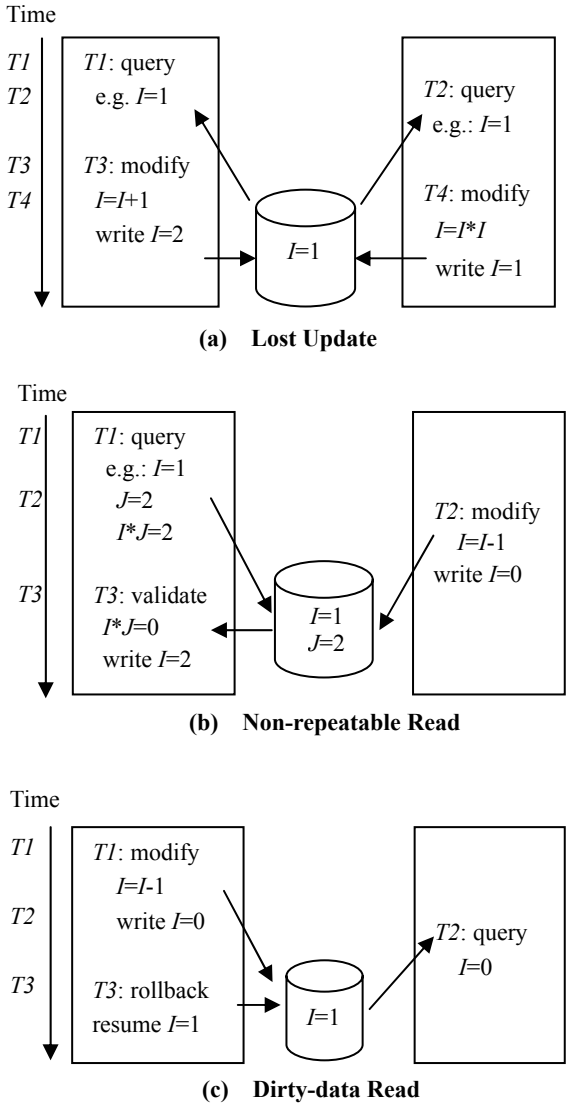


Figure 1 Inconsistency of Data

waiting for the permission of the lock request. But it cannot deal with the object that is waiting for the lock before you get the permission.

Optimistic Lock Strategy: This strategy is similar to the optimistic serializability method to some extent. It presumes that all of lock requests will be permitted. It is supposed that the request can be permitted after each of demander sends a lock request (called as tentative lock). The demander deals with the object based on the hypothesis. The process can be finished naturally; otherwise, the demander must repeal the process and restore to the original state by using the 'undo' operation.

3. 3-LEVEL LOCK PROTOCOLS FOR ENSURING DATA CONSISTENCY

3-level lock protocols in this paper are used respectively to solve the data consistency problem in different extent. The definitions of the exclusive-lock and the shared- lock used in the 3-level protocols is as following.

Exclusive-Lock: If the transaction T setups an exclusive-lock to the data object A, no other transaction can read or update A except T before T releases the exclusive-lock.

Shared-Lock: If the transaction T setups a shared- lock to the data object A, other transactions are only permitted to read A, but they could not be permitted to update A before T releases the shared-lock.

1-Level Lock Protocol: A transaction T needs to add an exclusive-lock to an operated object before modifying it, and releases the exclusive-lock at the end of process of the transaction. This protocol can avoid the problem of lost update.

2-Level Lock Protocol: A transaction T needs to add an exclusive-lock to an operated object before modifying it, and releases the exclusive-lock at the end of the transaction. Moreover, the transaction T must add a shared-lock to the operated object before reading it, and releases the shared-lock after reading. This protocol can avoid the problem of both lost update and dirty-data read.

3-Level Lock Protocol: A transaction T needs to add an exclusive-lock to an operated object before modifying it, and releases the exclusive-lock at the end of the transaction. Moreover, the transaction T must add a shared-lock to the operated object before reading it, and releases the shared-lock at the end of the transaction. This protocol can avoid all problems that include lost update, non-repeatable read and dirty-data read.

4. TRAFFIC LIGHT MODEL USED AS A NEW CONCURRENCY CONTROL MECHANISM

It can be known from above three kinds of protocols that they could not solve how to control concurrent operations when multi-operations arrive at the same time; they could not deal with the priority of reading and writing operation, etc.
Based on above protocols, we propose a new type of

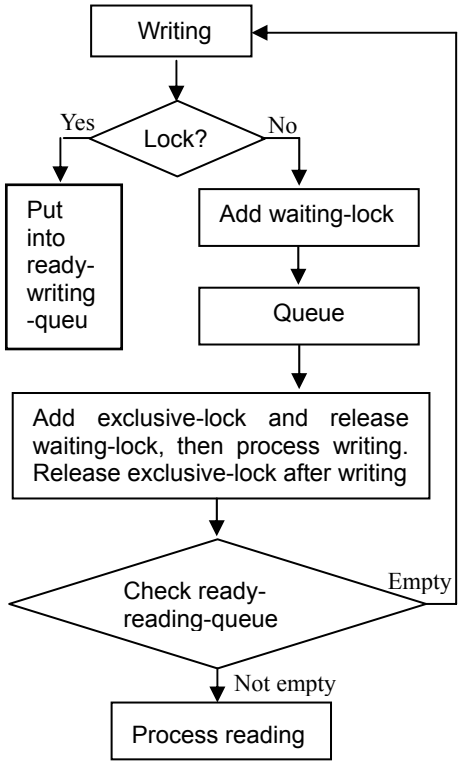


Figure 2 Process of Writing-Operation

concurrency control mechanism -- traffic light model. It is a concurrency control model that is similar to the traffic control at the crossroad. We add a waiting-lock (similar to the yellow light) besides the exclusive-lock (similar to the red light) and shared-lock (similar to the green light). This model defines some basic rules as following:

① The elementary unit which the model processes is an operation such as reading, writing, and so on.

② We assume the priority of a reading-operation (similar to a foot-passenger) prior to that of a writing- operation (similar to a car) because real-time demands of reading-operation generally is higher than that of writing-operation and the executing time of reading- operating is less than that of writing-operation.

③ The exclusive-lock (similar to red light) shows that the writing-operation is carrying on.

④ The shared-lock (similar to green light) shows that the reading-operation is carrying on.

⑤ The waiting-lock (similar to yellow light) shows that there are some operations waiting for the completion of current executed operation.

⑥ There is not any lock acting on data object, which means idle state of object.

⑦ The exclusive-lock (similar to red light) can not be active when the shared-lock (similar to green light) is active, and vice versa.

⑧ When multi-writing-operations are requested at the same time, they should queue and be put into a ready-writing-queue.

⑨ When multi-reading-operations are requested at the same time, they should queue and be put into a ready-reading-queue.

Figure 2 shows that when a writing-operation arrives, it should be processed as following.

① It should be put into ready-writing-queue if its operated object have been locked by some lock such as writing, reading or waiting lock.

② If the state of its operated object is idle, the waiting-lock should be firstly setup on the object, and then more then one writing operations need be queue. Subsequently, an exclusive-lock should be added and the waiting-lock should be released, and then an operation is chosen to run from the ready writing queue.

③ The exclusive-lock should be released and the ready-reading-queue is checked after completing the writing operation. If there is some reading operation in the queue, it should be processed in the first place, otherwise, the writing operation is checked.

When a reading operation arrives, it should be processed as shown in Figure 3. The executing process is following.

① It will be put into ready-reading-queue and the waiting-lock be added if its operated object have been locked by the exclusive-lock.

② Otherwise, the waiting-lock should be firstly added on its operated object. Then if there are more then one reading operations, they should be queued. Subsequently, a shared-lock should be added and the waiting-lock should be released, and then an operation should be chosen to run from ready reading queue.

③ The shared-lock should be released and ready-reading-queue is checked after carrying out the writing operation. If there is some reading operation in the queue, it should be processed in the first place, otherwise the writing operation should be checked.

When the reading operation and the writing operation arrive at the same time, the reading operation should be processed firstly.

5. EXAMPLE

We presume there are several reading operations and writing operations in a transaction and the writing operations must spend more time and occupy more resource than the reading operations. Because the traditional concurrency control mechanism does not deal with the priority of reading and writing operation and its elementary unit is a transaction, the writing operation will probably be processed firstly, which will result in delaying reading operations and losing the real-time property. However, these problems can be solved easily and efficiently by means of our proposed method of traffic light model.

6. CONCLUSION

It can be known from our description about traffic light model: it can solve data consistency problem in the concurrency control; and this model can process the priority of reading and writing operation. Compared with simple 3-levels lock protocols, this model is easier to be understood and more efficient in access control aspect. The elementary unit that the model processes is not transaction but operation, so it can be more convenient and precise to control the users' operations in the database system.

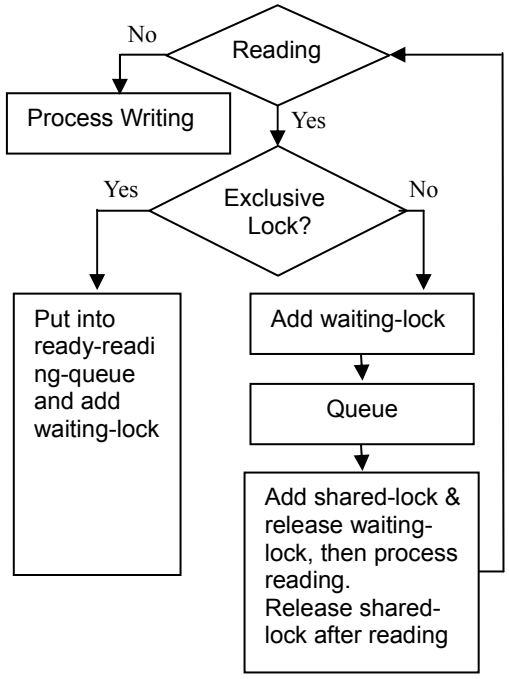


Figure 3 Process of Reading Operation

7. REFERENCES

[1] S. Wang, H. Chen, Tutorial of Database System Theory, Tsinghua University Publishing, Jul. 1998: 145-173

[2] M. L. Shi, Y. Xiang, G.X. Yang. The Theory of CSCW And Its Applications, 2000

[3] C. A. Ellis, S. J. Gibbs, G. L. Rein, "Groupware: Some Issues and Experiences", Communication of ACM, 1991

[4] J. Muson, P. Dewan. “A Concurrency Control Framework for Collaborative Systems”, Proc of ACM Conf on Computer Supported Cooperative Work, 1996
[5] M. Suleiman, et al, “Serialization of Concurrent Operations in a Distributed Collaborative Environment”, Proc of ACM SIGGROUP on Supporting Group Work, 1997
[6] C. Sun, C. A. Ellis. “Operational Transformation in Real-time Group Editors: Issues, Algorithms, and Achievements”, Proc of ACM Conf on Computer Supported Cooperative Work, 1998
[7] C. Sun, X.H. Jia, et al, “A Generic Operation Transformation Scheme for Consistency Maintenance in Real-time Cooperative Editing Systems”, Proc of ACM SIGGROUP Conf on Supporting Group Work, 1997
[8] G.X. Yang, M.L. Shi. “Programming Language Oriented Cooperative Application ---- Cova”, Chinese Science, 2000
[9] G.X. Yang. “Subgroup Ware ---- Cova Programming Language and System Research”, PhD Thesis, TsingHua University, 2000

Specific Features Of Information System (Is) Development For Distributed Database Environment With Client/Server Architecture

Hilaire Nkunzimana ,M.SC
Yao Lin Gu, Professor
E-mail: hilaire75@hotmail.com

ABSTRACT

The purpose of this paper is to complement standard steps of IS development with the specifications that are necessary for the applications being created in the environment of distributed databases with the client/server architecture.

Keywords: Client/server model, Distributed database environment, MDIS

1. INTRODUCTION

Demands on increase of accessible, fast available and safety information and operational cost of system reduction are impulses for designer of IS in development new systems based on distributed data processing. In our paper we specify one way to reach this goal, i.e. IS implementation in distributed database environment with client/server architecture.

2. METHODOLOGY MDIS ("MULTIDIMENSIONAL DEVELOPMENT OF INFORMATION SYSTEM") - OVERVIEW

The Multidimensional Development of Information Systems (MDIS) methodology has been developed at the University of Economics in Prague. The dimensions of IS lifecycle are of three types. The first is the time dimension (the positions in time dimension correspond to the phases in the IS lifecycle and the levels of abstraction used in IS design). The second type, the design dimensions, concerns the area or aspect of the IS being developed. The third type, the management dimensions, concerns the other aspects of IS development and operation such as methodological and management issues. We use MDIS to define standard phases of IS lifecycle.

3. CLIENT/SERVER ARCHITECTURE MODELS

The dominant model for client/server computing was developed and published by The Gartner Group. Application is divided into three logical parts - presentation, business logic and data management, and two physical parts - client and server (in context of our paper client and database server). This model is denoted usually as "two-tier architecture model". The model proposes five ways, how to map logical parts to physical parts: distributed presentation, remote presentation, distributed function, remote data management and distributed data management. In context our paper, the fifth design in the standard model, distributed database, splits the database itself. This design is different from others in two ways. First, it describes distributed data, while the others describe distributed processing. Second, distributed data is usually managed by distributed database system; consequently, it's completely transparent to the programmers. The others are not .
Another model client/server computing is denoted as "three-tier architecture model". This model is based on two-tier model in dividing application into three logical parts. Physical parts is divided into client and n-servers. N-servers are, as a rule, application server and resource managers such as RDBMS (Relation DBMS) or DDBMS (Distributed DBMS). In this model we prefer to use the term presentation logic (physically clients) and application logic (physically application server which contains most of the business logic) and data logic (physically resource managers) for denotation of physical devices. Application server and resource manager can be located in one physical device. In implementation of this model transaction monitors as Transarc's Encina or USL's Tuxedo monitors are typical used as application server.
In this section we compare two-tier and three-tier architecture model. The problem of model is significantly reduced into data logic, which is theoretically completely removed in three-tier model from the client. In picture 2 we compare two-tier a three-tier architecture.
If the data schema is changed in the two-tier architecture, then clients cannot operate, because each client must have data logic.
If the data schema is changed in the three-tier architecture, there is only one breakage point. Another benefit of three-tier model is that the data logic doesn't need to be ported to each client platform, but the draw-back is that there are more components to maintain in system with more communication links. Another problem of three-tier model is that it requires more forethought than two-tier architecture (i.e. dividing application into presentation logic/services, business logic, data logic/services), but the complexity is added to transaction management, which solely resides on the application server. The three-tier architecture adds more components to the network, thus increases network failure. But the real purpose of the three-tier architecture is to overcome heterogeneity - it hides the source of the data by removing data logic from the client and splitting it between application and database server.

4. DATABASE MANAGEMENT SYSTEMS

4.1. Basic architecture of databases in distributed system

There are three common basic architecture of databases on distributed system. These are the following:
independent databases,i.e. autonomous DBMS in various nodes of distributed system, with no data integration. Distributed transaction, if needed, are performed by application program in client node or in application server
federated databases, i.e. loose cooperation of independent databases. Partial data integration and cooperation is provided by global rules (e.g. naming conventions, import/export mechanisms) which are obeyed by all DBMS. There are still no GCS (Global Conceptual Scheme) and distributed transactions/operations are carried out by application program (client node or application server).
distributed databases. In this architecture the application program is provided with logical view of one logically centralized database. There is (one) GCS, providing high data

integration. At the same time the autonomy of DBMS in various nodes is very limited. Distributed operations are carried out by DBMS (or by application program in client node in case the DBMS is not installed there). For this architecture there are three ways of data allocation, which are controlled by database system. These are the following:
partitioned database - nonreplicated database contains fragments (horizontally or vertically divided relation) that are allocated to sites, and there is only one copy of any fragment on the network, fully replicated database - the database exists in its entirety at each site. We can define two models, with or without master copy, partially replicated database - fragments are distributed to the sites in such a way that copies of fragment may reside in multiple sites.

4.2. Relationship between client/server architectures and architecture of databases

In case of independent or federated databases we prefer to implement the three-tier architecture model of client/server computing. Application server (business logic) is coordinator of local database schemes and transaction monitor which is implemented at this level, executes on-line transaction processing.
In case of distributed databases we prefer using the two-tier architecture model of client/server computing. Business logic can be implemented as stored procedure, rule or trigger (remote presentation or distributed function models of two-tier architecture model). On-line transaction processing is executed by distributed databases.

5. DESIGN OF IS

In this part of paper the design of IS - with emphasis on client/server architecture and distribution problems - is covered. We specify some steps and questions concerning the development of IS with given characteristics. Steps and questions are defined for data, function, hardware and software dimensions for all phases of the MDIS framework.

Phase: Business strategies
Specification of the processing type in company, i.e. data-oriented, process-oriented, socio-technically oriented, real-time system, distributed system, etc.
Trends in company finding - business needs of company - the need to centralize or decentralize processing; rightsizing and downsizing, etc.
Determining geographical locations of company (present and future) - various types of location, national or international placement of location and after that mapping of these results to organization chart

Phase: Information strategies
Finding the level of support for enterprise's activities by information technologies in each location of company and plans to future technological support.
The definition of functional areas (subsystems) is based on classes of related data and main functions.
Specification of relationships between locations of company and functional areas (subsystems).

Phase: Feasibility study
Determining communication links between various locations of company.
Specification of basic technological kernel of system, i.e. hardware resources (servers, workstations, communication device), software resources (operating systems, DBMS, communication software and data access technologies) and client technological software. Answers to following questions can help in decision on which DBMS to choose:
Should we distribute data, or do we need to distribute only the access to data with the data itself remaining at the site.
Do we have data administrator? The database administration function needs more sophistication in DDBMS environment than in centralized database environment.
Do we want ultimate authority over data definition and access handled locally, globally or both? What type of local autonomy needs to be implemented in DBMS?
When a communications line fails, how do we balance the need for availability with the need for data integrity? How do we balance the cost of recovery with the need for integrity?
In updating replicated data, how do we weigh the cost of simultaneous updates vs. the need for data integrity?
Selection of part of system which will be implemented.

Phase: Global design
Specification of functional areas interface in relation to global data model.
Requirements on data security in subsystem.
Decision on importance of having the latest copy of the data.

Phase: Detail design
Determining dominant and critical functions of system and detail specification of qualitative characteristic of IS (information needed for configuration of IS). Answers to following questions can help:
Maximum and minimum number of instances of each entity.
How many instances (maximum and minimum) of given entity are in each relationship?
How often are entities accessed by function?
How many instances of entity is required during one execution of function (enquiry and update)?
How often is data updated during a specific time interval (only for updated functions)?
Decision on importance of having the latest copy of the data for these functions?
Specification of enquiry and updated access path to data for these functions.
For data replication the approach with or without master copy is specified. In case of replication with master copy a time interval or other mechanisms for update of copies is defined. In case of replication without master copy, steps and direction of update are defined also.
Decision concerning backup and recovery strategies (technological aspect).
Making up safety plan and defining basic safety rules for hardware, network, data and function level.
Detail specification of technological kernel of system. Specification includes decision on quantitative characteristics (transfer rate of network, disk and memory capacity, etc.) and qualitative characteristics (heterogenous environment DBMS, access to distributed data, communication protocols, etc.), for:
nodes (servers) and workstations - solution of problems concerning the configuration of nodes (servers) and placement of nodes and workstations in locations. Defining capacity and performance of devices.
network - solution of problems concerning the modelling of computer network (i.e. computer network design - backbone, LAN and WAN, optimalization of LANs interconnection, allocation of redundant link for the increasing reliability of computer network) based on heuristic methods or operating research methods.
DBMS - specification of the basic characteristic of DBMS (data access - remote unit of work, distributed unit of work or

distributed request, fragmentation, replication, etc.).
Design of logical data model (relations, integrity constrains, primary and foreign keys definition, index definition, etc.) for selected environment.
Selection of client/server architecture model based on the technological kernel of the system and the exact specification of the program system, i.e. dividing of the application in two or three levels.

Phase: Implementation
Based on technological kernel of system (it is restricting element now), logical data model and model of program system, quantitative and qualitative information, the implementation of data model into physical realization (physical tables, fragments, copies) and the implementation of programs into client server architecture (relationship between client/server computing and architecture of databases in distributed system).
Based on strategies of data processing (minimalization of response time vs minimalization of operational cost) allocation of resources - data and application.
New allocation of data and applications based on quantitative and qualitative information from testing.
Note: If prototyping is used, some decision in implementation phase are mixed with decisions in detail design phase.
Phases: Installation and operation and maintenance
New allocation of data and applications based on quantitative and qualitative information from operation (response time, operational cost) or based on changes in company (new dislocation of company or functions, etc.)

6. CONCLUSION

As was shown above in our paper, the following major problems in design of IS in distributed database environment with client/server architecture were specified:
Combination of client/server architecture models with architecture of databases.
"Optimal" division of the application into presentation, business and data logic and placement of these parts at client and server (application and data server)
"Optimal" placement of data and applications in system.

7. REFERENCES

[1] Burleson,D., Managing Distributed Databases: An Enterprise View, Database Programming & Design, June 1994 b
[2] Ga,L.,JandoJ. On the Access to Distributed Data and it's Standardization, conference CONnectivity 1994, Hagenberg, Austria
[3] Kiernan,C., Client/Server: Learning from History, Database Programming & Design, September 1993
[4] Loosley,Ch.,Look Before You Leap, Database Programming & Design, September 1993
[5] Riha,K.,Vo_ek,J., Design and Management Dimmension of Large Information Systems - The "Czech/Lands" Perspective, Conference on the Theory, Use and Integrative Aspects of IS Methodologies, September 1993, Edinburg, UK
[6] su,M.T.,Valduriez,P., Principles of Distributed Database Systems, Prentice-Hall,Inc., 1991
[7] Winsberg, P., Designing for Client/Server, Database Programming & Design, July 1993
[8] G.Coulouris,J.Dollimore,T.Kindberg,Distributed systems 3rd edition,Addison-Wesley,2000.

Application of the Parallel Accelerating Board Based on ADSP-21062*

Gao Shu Guo Qingping Gao Jie
Computer College ,Wuhan University of Technology ,Wuhan 430063 P.R.C
E-mail: gshu418@21cn.com

ABSTRACT

Firstly, the paper introduces the structure and working principle of the ADSP-21062, the character of the structure and the programming of the parallel accelerating board .Then , taking non-linear transient equation in the temperature field and the Mandelbrot set problem for example, it puts emphasis on the application of the board in the large-scale numeric and non-numeric computation problem. The testing result shows the board is a kind of high-performance parallel processing product , it can be widely used in the fields such as national defense scientific research ,intelligence process and analysis ,simulated training and simulation ,real-time industrial control .

Keywords:ADSP-21062 parallel algorithm Domain Decomposition

1. INTRODUCTION

The parallel accelerating boards, developed by the Wuhan Digital Engineering Institute, are board-level products based on the ADSP-21062 . They can be used as plug-in boards and inserted into the various mainframe environments(including Multibus ,ISA ,PCI ,Compact PCI,VME and other standard industrial buses).They can be flexibly configured to construct the high-performance parallel systems according to the need of user, and can provide the high-speed floating-point computing capabilities of several hundred million times per second to several ten billion times per second .Because these parallel systems ,which mainly consist of the boards and host computer ,have small volume , light weight , low power consumption ,fast computation speed ,they can be used in radar and sonar signal processing ,graphics and image processing ,high-speed real time control ,and mounted on vehicles, aircraft ,ships and crafts ,satellites and missile ,and can also be applied in the ground or underground operation under the adverse circumstance .

Therefore ,we study the capabilities of the parallel accelerating boards in solving the problem of large-scale non-numeric computation and large-scale numeric computation .The conclusion shows that their performance are stable ,the function powerful, the usage convenient ,computation speed fast .So they will be able to widely use in the fields which need the high speed and has the large amount of data to be calculated ,such as national defense scientific research ,intelligence process and analysis ,simulated training and simulation ,real-time industrial control .

2. THE CHARACTER OF THE STRUCTURE AND THE PROGRAMMING OF THE PARALLEL ACCELERATING

2.1 ADSP-21062

The ADSP-21062 SHARC—Super Harvard Architecture Computer—is a high performance 32-bit digital signal processor for speech, sound, graphics ,and imaging applications. Its block diagram is showed in the figure 1.

It includes:

- There are three on-chip buses of the ADSP-21062: PM bus (program memory), DM bus (data memory), and I/O bus.
- Core processor :the core processor of the ADSP-21062 consists of three computation units, a program sequencer, two data address generators, timer ,instruction cache, and data register file.

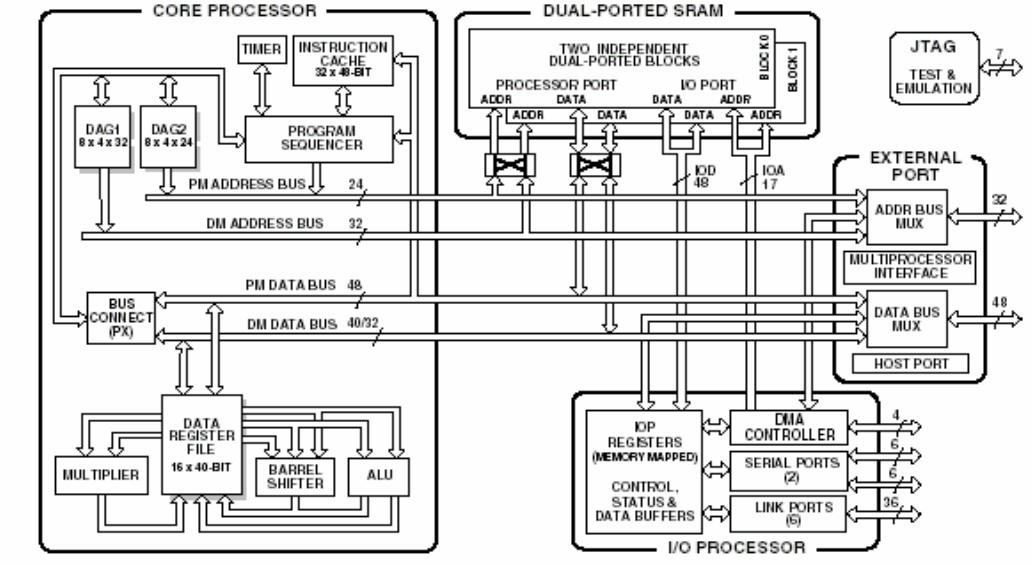


Figure 1 ADSP-21062 Diagram Block[1]

*Supported by Natural Science Foundation of China(No.60173046.)

● Dual-Ported Internal Memory: The ADSP-21062 includes a 2 Mbit SRAM, organized as two 1 Mbit blocks. Each memory block is dual-ported for single-cycle, independent accesses by the core processor and I/O processor or DMA controller. The dual-ported memory and separate on-chip buses allow two data transfers from the core and one from I/O, all in a single cycle.

● External Memory & Peripherals Interface: The ADSP-21062's external port provides the processor's interface to off-chip memory and peripherals. The 4-gigaword off-chip address space is included in the ADSP-21062's unified address space. The separate on-chip buses—for PM addresses, PM data, DM addresses, DM data, I/O addresses, and I/O data—are multiplexed at the external port to create an external system bus with a single 32-bit address bus and a single 48-bit data bus.

● Host Processor Interface :The ADSP-21062's host interface allows easy connection to standard microprocessor buses, both 16-bit and 32-bit, with little additional hardware required. The host interface is accessed through the ADSP-21062's external port and is memory-mapped into the unified address space. The host can directly read and write the internal memory of the ADSP-21062, and can access the DMA channel setup and mailbox registers. Vector interrupt support is provided for efficient execution of host commands.

● Multiprocessing: The ADSP-21062 offers powerful features tailored to multiprocessing DSP systems. The unified address space allows direct interprocessor accesses of each ADSP-21062's internal memory. Distributed bus arbitration logic is included on-chip for simple, glueless connection of systems containing up to six ADSP-21062's and a host processor. Maximum throughput for interprocessor data transfer is 240 Mbytes/sec over the link ports or external port. [2]

● I/O Processor: The ADSP-21062's I/O Processor (IOP) includes two serial ports, six 4-bit link ports, and a DMA controller.

✧ Serial Ports :The ADSP-21062 features two synchronous serial ports that provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices. The serial ports can operate at the full clock rate of the processor, providing each with a maximum data rate of 40 Mbit/s. Independent transmit and receive functions provide greater flexibility for serial communications.

✧ Link Ports; The ADSP-21062 feature six 4-bit link ports that provide additional I/O capabilities. The link ports can be clocked twice per cycle, allowing each to transfer 8 bits per cycle. The link ports can operate independently and simultaneously, with a maximum data throughput of 240 Mbytes/s. Link port data is packed into 32-bit or 48-bit words, and can be directly read by the core processor or DMA-transferred to on-chip memory.

✧ DMA Controller: the ADSP-21062's on-chip DMA controller allows zero-overhead data transfers without processor intervention. The DMA controller operates independently and invisibly to the processor core, allowing DMA operations to occur while the core is simultaneously executing its program. Both code and data can be downloaded to the ADSP-21062 using DMA transfers .

2.2 The character of the structure of the parallel accelerating board

There are four Analog Devices ADSP-21062 SHARC DSP chips on a parallel accelerating board, which are connected to form a Multiprocessing cluster through a shared bus. It can achieve the computing power up to 480MFLOPS .It uses the six link ports on the ADSP-21062 to support point- to- point communication ,and provides a shared bus by means of the distributed bus arbitration logic included on the chip . It also has 1M×32bit zero waiting shared SRAM that is used as external shared memory. At the same time, in order to support system extensibility ,the board provides 12 external link ports ,each of which is 40Mbyte/sec,and 2 serial ports of TDM mode, each of which is 40 Mbits/sec. There is a JTAG interface which is used by the EZ-ICE simulator and a 512k × 8bit Flash ROM which is to support the modular electrifying bootstrap. Its structure block diagram is shown in the figure 2.

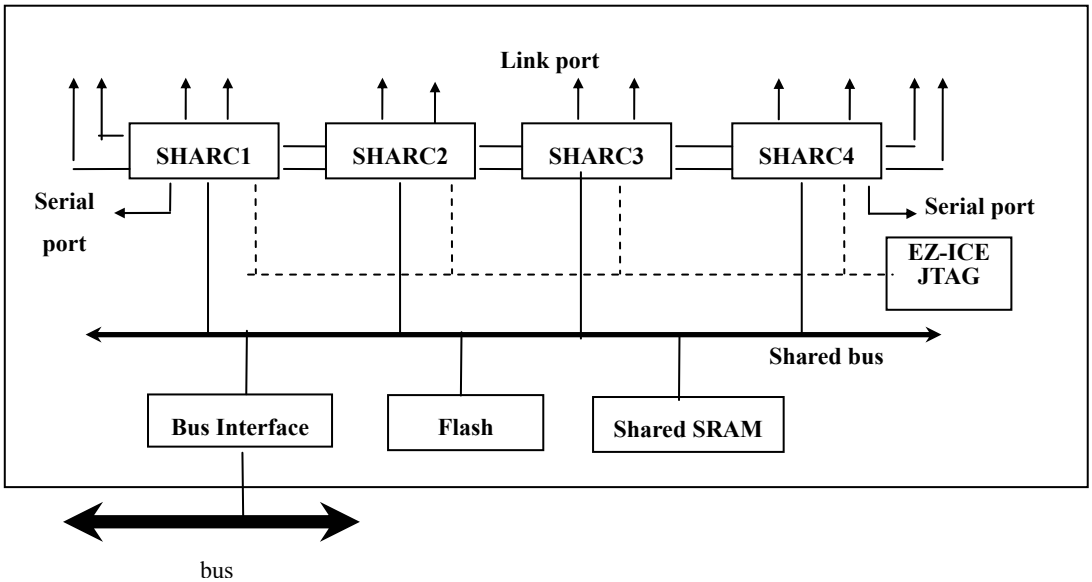


Figure 2 The structure block diagram of accelerating board

2.3 The character of the programming of the parallel accelerating board

Generally, in the parallel programming ,the computation objects are divided into a group of related function modules according to their function ,and these function modules are distributed to the most appropriate processors.

According to the character of the parallel accelerating board ,the following methods can be adopted by the parallel

algorithm :

(1) Master-slave decomposition method.

The kind of technology decomposes the application problem into a lot of tasks ,in which one is a master task, the others are slave tasks. The master ,which runs on the main processor, is responsible for managing the whole data structure ,maintaining the task table and assigning the jobs to the slaves .At the same time, the slaves ,which run on the different processors, are used to perform various calculation.

(2) Definition domain decomposition method.

It divides the data definition domain of the problem into a lot of the parallel tasks ,and restricts that every task can but updating its sub-domain.

Besides , the communication mode is very important in the parallel programming .In the board ,there are several modes as follows.

● The mode of link port. It is a effective one to realize the communication among the ADSP-21062s .Each ADSP-21062 has six link ports which can implement the point-to-point communication with other five ADSP-21062s simultaneously .It uses the on-chip DMA to send and receive the data whose speed is 80MB/S.However ,every time ,only 4 bits data can be put into the data line .

● The mode of external port .The mode is more flexible .It can only set the sender or receiver. Moreover ,it can communicate between the SHARC and host processor ,and also between the ADSP-21062s. Because all of the external ports share the bus ,its speed is up to 240MB/S.However ,its limitation is the communication have to be complete serially .At the same time ,it has the data-packed ability ,namely, it can pack the 32/48 word data on the chip into 16/32 word ones in order to be accessed by the host processor or the peripherals.

● The mode of direct access to bus. It means the holder of the bus can access the multiprocessor memory space and external memory space of the other chips. For example , because the address of the registers may be mapped into the IOP space and are known and fixed ,the general registers MSGR0-MSGR7 can be used to transfer data among the ADSP-21062s .If only both ADSP-21062s ,which want to communicate, appoint the used register and the one put the data into the register ,the other can get them. Similarly, by means of the mode ,ADSP-21062 or host processor can also write directly the address of the interrupt service routine to vectored interrupt register on the other ADSP-21062 in order to make the vectored interrupt to be generated on that ADSP-21062 and execute the routine to control the other ADSP-21062.

In consideration of the above characters ,we study the large scale numeric and non-numeric computation problems respectively . For the former , we take the heat conduction model of the ceramic/metal composition material as example to research the high-efficient parallel algorithm for the large-scale and complicated heat problems . For the later ,we take the famous Mandelbrot Set as example. We optimize it by means of the method of distributing coordinates in the space alternately to maintain the load balancing on every ADSP-21062.

3. AN APPLICATION OF THE BOARD IN LARGE-SCALE NUMERIC COMPUTATION PROBLEMS

3.1 The description of the problem

We consider a non-linear heat transfer equation of ceramic/metal composite material:

$$\mu_1 = \frac{\partial}{\partial x}(k\frac{\partial u}{\partial x}) \qquad 0 < x < 5; t > 0 \tag{1}$$

here boundary conditions are

$$\mu(0,t) = \cos(2\pi t) \qquad \mu(5,t) = 0$$

and initial condition is

$$\mu(x,0) = 0 \qquad (0 < x < 5)$$

equation(1) may be rewritten as

$$\mu_t = ku_{xx} + k_x\mu_x$$

$$\text{or} \quad \mu_t = k\mu_{xx} + G \tag{2}$$

$$\text{here} \quad G = G(x,t) = k_x\mu_x$$

and

$$k = (1 - f_m)k_c + f_m k_m$$

$$k_c = 0.1 - 0.01\mu + 0.001\mu^2$$

$$k_m = 1 + 0.1\mu + 0.01\mu^2$$

$$f_m = \begin{cases} x^2 & x \le 1 \\ 1 & x > 1 \end{cases}$$

where the k_c represents feature of the ceramic ,and the k_m feature of the metal in ceramic/metal composite material. The equation (2) is a non-linear equation since k is a function of μ .

3.2 The research and implementation of the multi-grid parallel algorithm with VBF(Virtual Boundary Forecast) of the non-linear transient equation in the temperature field

In order to solve the non-linear transient equation ,we adopt the multi-grid parallel algorithm with VBF , proposed by Guo Qingping in the reference [3],but following technique should be also added to avoid that the computation does not converge or converge falsely and speed up the computation:

(1) The false convergence means that the computation should have converged according to the error ,but in fact it does not yet. Therefore , a constant Nmin is defines as the minimum loop number in the every sub-domain computation .Only when the loop number is greater than the constant can the computation be terminated.

(2) In order to speed up the computation, we take two steps to solve the problem .Firstly ,we calculate the approximate initial values on a very coarse grid ,whose step is very large .The values are not necessary for precise values ,just as the initial guess values for virtual boundary. Its goal is to get the good initial values , because the approach can transfer the information of the known boundary quickly. In this stage, every computer performs the same computation ,gets the same initial results, then takes out the results which are related to its sub-domain .

The algorithm is based on the domain decomposition method .It forecasts the virtual boundary values (the virtual boundary are produced by the domain decomposition method),and uses the multi-grid method. All of these quicken the convergence , reduce the communication ,and improve the speedup.

In realizing the algorithm on the parallel system based on DSP, the whole domain is divided into some sub-domains according to the number of ADSP-21062s .Each sub-domain has its boundary and its initial values .Every ADSP-21062 calculate a sub-domain independently .ADSP-21062s exchange the boundary data by means of the link port. Link port is especially useful for point-to-point communication in

multiprocessing systems. Each link port has its own double-buffered input and output registers. Clock/acknowledge handshaking controls link port transfers. Transfers are programmable as either transmit or receive.
Firstly all ADSP-21062s calculate the initial values on the very coarse grid .Secondly ,the host processor send the computation task to each ADSP-21062 ,then ADSP-21062s perform the task. In needed time, ADSP-21062 can communicate with the neighbor ADSP-21062 to exchange the data on the boundary. After computing ten time steps ,every ADSP-21062 send the results to the host processor to draw the graph.

3.3 Testing result

We insert the two parallel accelerating boards into a kind of military computer ,which includes Multibus , to construct the parallel system .For the problem of non-linear heat transfer equation, the testing result is as follows.

Number of sharc	Runtime(second)	Clock number	Speedup	Parallel efficiency
1	56.95(s)	1037	1.00	1.000
2	28.62(s)	521	1.990	0.995
3	19.13(s)	348	2.980	0.993
4	14.56(s)	265	3.913	0.978

We also test the result only using the host processor to do the same job.

CPU	Running time(second)	Clock number
80486DX2/66	390.12	7103

4. AN APPLICATION OF THE BOARD IN THE LARGE-SCALE NON-NUMERIC COMPUTATION PROBLEM

4.1 The description of the problem

Mandelbrot Set is the set M which is composed of the following complex number:

$$M = \{C :| M_n(c) |< \infty, \forall n \in N\}$$
$$where: \ M_0(c) = 0$$
$$M_{n+1}(c) = M_n(c)^2 + c$$

It can also be expressed as :

$$\text{if } \exists n :| M_n(c) |> 2 \quad \text{then} \quad c \notin M$$

Because the complex number can be represented by the coordinate of the point in the complex plane ,the set can be drawn on the display .The method is that if a complex number is not in the set ,the color value of the corresponding point is the loop number in the program which decides the number is not in the set ,and if the loop number is more than a fixed value and the module of the complex number is still less than 2,it is considered that the complex number is in the set and the color value of the point is set black ,that is to say, it is zero. It can be proved that all of the complex numbers represented by the points in the square are in the set if the complex numbers represented by the points at the edge of a square are all in the set .

4.2 The realization of the Mandelbrot Set

(1)The design of the algorithm
In order to draw the picture of Mandelbrot Set ,it is necessary to calculate the color value of every point in the complex plane .As we all know ,the calculation is independent. Therefore, the master-slave decomposition method can be used.
With regard to the mode of the communication, the data transmission between the ADSP-21062 and host processor is by means of the mode of external port ,and the mode of the direct access to bus is also used to set up simple handshaking control by way of the general registers MSRG6.
The parallel program consists of the master.c and newslave.c ,in which the master.c is the main process and runs on the host processor ,and the newslave.c is the sub-process and runs on each ADSP-21062 . The main process is responsible for loading the newslave.ldr to the every ADSP-21062 , initiating the data and sending them to the each sub-process by means of the sendsharc function ,receiving the calculation results from the every sub-process by means of the receivesharc function and drawing the picture. Meanwhile, each sub-process receives the task by the receive function ,calculates the color value of points and sends the results to the main process by the send function ,and getting the new task from it.
(2)The method of implementation
In order to reduce the communication between the main and sub process, we use the method of distributing coordinates in the space alternately to maintain the load balancing on every ADSP-21062 .It means the complex plane will be divided into x_block*y_block blocks ,which are represent by coordinate of the point at most left ,and a block will be as a computed task .In order to prevent some sub-process from being the bottle-neck of the whole computation ,when the main process waits for the results ,we adopt the polling scheme rather than sequential one to test which ADSP-21062 will send the result. At the same time ,a address table(namely config.dat file) is set up to store the address of the parallel accelerating board and its ADSP-21062s in order to make the program support expansion. When the program runs ,it read the address to CGF array from file in order to provide the address of the board and ADSP-21062s .If a board is added into the system, we only need to add its address to that file ;if a ADSP-21062 is wrong ,we only need to delete its address from that file .All of these do not influence the execution of the program.
4.3 Testing result
At the same time ,using 1~8 ADSP-21062 on the same parallel system as above application to draw a same picture of Mandelbrot Set respectively ,we test the results of the runtime, clock number ,and the calculated speedup, parallel efficiency as followed.

Number of sharc	Runtime(second)	Clock number	Speedup	Parallel efficiency
1	197.18s	3590	1.000	1.000
2	98.92s	1801	1.993	0.997
3	66.40s	1209	2.969	0.990
4	50.14s	913	3.932	0.983
5	40.59s	739	4.858	0.972
6	34.05s	620	5.790	0.965
7	29.72s	541	6.636	0.948
8	25.92s	472	7.606	0.951

If the same picture of Mandelbrot Set is drawn on the host processor ,the used time and clock number is as follows:

CPU	Running time(second)	Clock number
80486DX2/66	1356.33 s	24694

5. CONCLUSIONS

The above two application respectively use the bus and link ports to communication ,so they can reflect the performance of the parallel accelerating boards .After analyzing above result ,we can draw the conclusion that if boards are inserted into some systems, even if in which the speed of host processor is very slow, these systems can also gain very high parallel processing ability and finish the large-scale numeric and non-numeric computation .And the boards can adapt to a lot of buses ,such as Multibus ,ISA ,PCI ,Compact PCI,VME and other standard industrial buses, and has extensibility and stability ,so they are used to construct the parallel system based DSP ,which can be widely applied in the national defense scientific research ,intelligence process and analysis ,simulatedtraining and simulation ,real-time industrial control .

6. REFERENCES

[1] Analog Devices, Inc.ADSP-21062 C Runtime Library Manual，Third Edition，8/1997

[2] Analog Devices, Inc.SHARC User's Manual，Second Edition，7/1997

[3] Guo Qingping et al. 'Optimum Tactics of Parallel Multi-grid Algorithm with Virtual Boundary Forecast (VBF) Method Running on a Local Network with the PVM Platform' Journal of Computer Science and Technology, Science Press China and Allenton Press INC, USA,July 2000(4)

Analysis And Design Of The Software System For Voyage Data Recorder

Jian-hai Jin
College of Information and engineering, Yangzi University,
Wuxi, Jiangsu, 214036, China
E-mail: jjh3306@sina.com
And
He Ling
Software engineering department of China Ship Science Research Center
Wuxi, Jiangsu, 214082, China
E-mail: hihi@cssrc.com.cn
And
Wen-hao Leng
Software engineering department of China Ship Science Research Center
Wuxi, Jiangsu, 214082, China
E-mail: lengwh@cssrc.com.cn

ABSTRACT

In this paper, we introduce the system of VDR (Voyage Data Recorder). Then the analysis of the system requirement is made. At the end, a simple design of the software system is presented.

Key words: VDR (Voyage Data Recorder), NMEA0183, Pre-Set Module

1. INTRODUCTION

VDR, commonly called marine black box, is a monitor product for ship. It is used to collect and record the **static data** and **dynamic data** of navigation, and save the recent data to a special storage device, which is designed to survive in the shipwreck. After the storage device is regained, the recorded data in it can be read and reappear in a special equipment. It will be very helpful to find out the cause of the shipwreck.
On November 27, 1997, the international maritime organization passed the No. A.861(20) resolution— the performance standard of Voyage Data Recorder (VDR) . As its appendix, the suggestion for performance standard of Voyage Data Recorder (VDR) is the basic standard to the ship equipped with VDR. The following is its main contents:
VDR is a complete system, which includes data interface of handling and coding, storage medium, power supply system, software and the other relative items.
The software must be able to record the dynamic data, such as the date, time, speed, water-depth, main alarm, radar data, people's dialogue, and so on.
Recently, many domestic and oversea companies have been paying attentions to develop VDR. And some of them have already tested their VDR system on the ship. The VDR system is related to a lot of knowledge, and the software system is its important part. Some analysis of the software system is presented in this paper, which is made in a project developed by CSSRC.

2. THE REQUIREMENT OF THE SOFTWARE SYSTEM

The software system is the important part of the VDR system. Besides the storage medium, the software system has the close relationship to the most items of the resolution suggested by International Maritime Organization. In general, the following task is what the software system needs to complete:

2.1 Record Data

The data that needs be recorded includes the static data and the dynamic data. The static data, such as the ship's name (NAME OF SHIP), is fixed for a certain ship, and should be inputted at any time. Whatever, the data can not be deleted or modified again after confirmation.
The dynamic data includes ship's operation data and status data. Both of them need to be collected, handled and saved. Usually, the software system should check whether there are operation, such as operator instruction given to the engine and rudder, per 3-5 seconds. The check-interval is depended on the condition of the ship. When an operation is ordered or done, all of the dynamic data include operation data should be collected and saved. If not, the system collected all data per 15-30seconds. The audio data should be collected in succession.
The system is capable of keeping all of the dynamic data during the latest period, such as 12 hours or 24 hours, etc. The outdated data will be deleted automatically.

2.2 Monitor Power Switch

The switch signal of power supply can be monitored with the software system. When catching the signal which indicates that main power supply is cut off, the system can generate an alarm signal, and began to time 2 hours. If not receiving the resume signal of the main power supply in time, the system finishes the work and exits. Otherwise, the program should clear the timer, record the signal and works as normal.

2.3 Alarm

When receiving one of the instances below, the system can record the signal and send out an alarm:
A. the signal that the main power supply cuts off;
B. syntax error in the received data;
C. overtime error in collecting the dynamic data.

2.4 Display Or Print The Real-Time Data

If needed, the real-time dynamic data of navigation can be displayed or printed by the system whit display equipment and printer.

2.5 Data Review

The data review software can be used to read and display the data in the storage medium which is put in the voyage data record data cabin. This part of software isn't included in the software system of VDR, so it's not been presented in this

paper.

3. THE DESIGN OF THE SOFTWARE SYSTEM

To meet the above requirement of the software system, the relative system model should be like what is showed in the figure1.

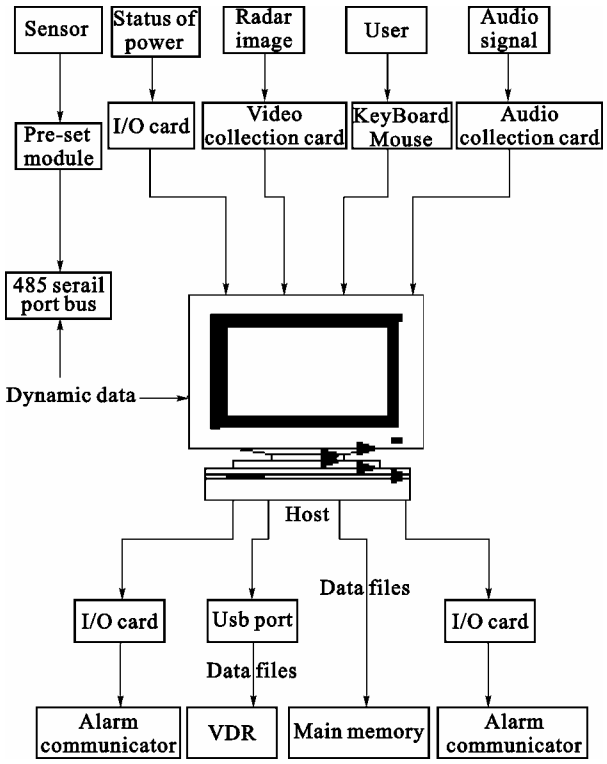


Figure1 hardware distribution and data flow

With our analysis, the software system could be divided into several modules: serial port communication module, NMEA0183 syntax handling module, audio handling module, image handling module, digital I/O module, data compression and storage module, alarming module and interactive module (as the figure2).

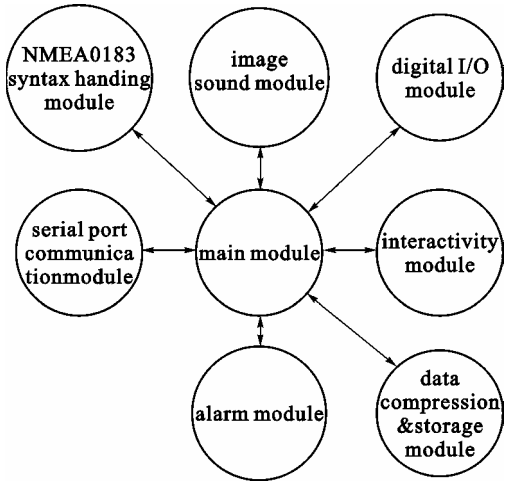


Figure 2 the modules of the software system

Following are the introductions of all these modules.

3.1 The NMEA0183 Syntax Handling Module

NMEA0183 is an industrial standard, which is established by American National Marine Electronics Association. NMEA0183 syntax handling is the conversation process of composing or decomposing the NMEA0183 sentence. The following provides a general explanation of the approved parametric NMEA0183 sentence structure:

$aaccc,x1, …xn*hh<CR><LF>

ASCII	HEX	DESCRIPTION
"$"	24	Start of Sentence
aaccc		Address Field. identifying the meanings of the data and Sentence Formatter
x_1, (…x_n)		Data field. its meaning is identified by the Address Field.
","	2C	Field delimiter. Starts each field except address and checksum field.
"*"	2A	Checksum Delimiter. Follows last data field of the sentence.
hh		Checksum Field.
<CR><LF>	0D 0A	Terminates Sentence.

The task of this module is to decompose or compose NMEA0183 sentence. The former procession is this: first step is checking whether the sentence is right, according to NMEA0183 protocol, then the sentence will be separate into some data fields. After that we get the useful data from the data field. On the contrary, the compose procession is united the useful data into fields, and generate a valid NMEA0183 sentence.

3.2 Serial Port Communication Module

In the design and implementation of the software system, an emphasis is how to get the dynamic data of navigation. On the ship, there are many different kinds of data. Except the audio and image data, the dynamic data could be divided into 2 sorts by data sources, which is depend on whether it is capability of outputting the NMEA0183 sentences or not. The data which doesn't satisfy the NMEA0183 protocol includes digit, on-off data and simulation data. For facility, all the data inputted into the main part of the software system should be formatted into the NMEA0183 sentences, and this task is done by pre-set module. Another task of the pre-set modules is buffering, which keeps the latest data. After pretreatment, all this data sentences are transferred in the serial medium.

The task of serial port communication module is to handle the communication in the serial medium between the host system and pre-set modules. With it, the host system can talk with pre-set modules. The communication between the host and pre-set modules is inquiry and reply.

The working process is as the following: the host sends the NMEA0183 inquiry sentences; the pre-set modules receive the inquiry sentence and made the parsing. With parsing, the pre-set modules decide which sentence in the buffer is needed, and consequently send a NMEA0183 reply sentence to the host. The host receives the reply sentence and deal with it. Then the next comes on.

3.3 Audio And Image Module

The function of audio module is to collect the sound both in the cab and in the broadcast by audio collection card.

The function of image module is to get the video information

including the radar by video collection card, and generate the image.

3.4 Digital I/O Module

The main function is to operate the I/O card. For the input part, it can monitor the status of the power supply. For the output part, it can send out the signal to alarm.

3.5 Data Compression And Storage Module

In general, the volume of the files of audio and image is very large. So the files need to be compressed. For example, we could convert the BMP files to JPG files. The storage includes the hard disk and the special storage medium. We can copy the data from hard disk to the recorder by USB. The data can also be written to the recorder directly by USB. When new files created, the outdate files will be covered by the rule of FIFO.

3.6 Alarm Module

The function of this part is simple. General alarm only needs to be displayed in the interface of the program. As serious alarm occurs, the signal can be sent to special alarming equipment through I/O card.

3.7 Human-Computer Interactivity Module

Function of this part is also quite simple. The current voyage data and the change of certain parameter during the late period of time can be displayed in the software interface.

4. CONCLUSION

This software system has been realized through VC++6.0 program language under windows2000 operation system, and now are prepared to equip in a ship for further test and improvement. This paper provides a brief introduction of the entire system software of the host software.

5. REFERENCES

[1] http://www.rycbgcgs.com/cbll/0512.htm [EB/OL]
[2] http://www.sz.chinanews.com.cn/no1_port/2001-02-17/2/2606.html[EB/OL]
[3] http://www.ankeer.com/web/zhuyaochanpin/wenzhang/vdr.htm [EB/OL]
[4] http://www.cmscyber.com/cmsweb/index.html [EB/OL]
[5] NMEA. NMEA 0183 standard For Interfacing Marine Electronic Devices (Version 3.00) [S] .July 1, 2000